Robust solvers for large indefinite systems in seismic inversion

Shanks, Douglas

Award date:
2015

Awarding institution:
University of Bath

Robust solvers for large indefinite systems in seismic inversion

J. Douglas Shanks

PhD

May 2015

Robust solvers for large indefinite systems in seismic inversion

submitted by J. Douglas Shanks

for the degree of PhD

of the University of Bath

Signature of Author .....

J. Douglas Shanks

ABSTRACT

In this thesis we study and develop iterative methods for solving the linear systems arising from discretisations of the Helmholtz equation. The Helmholtz equation is the simplest model used to describe high frequency wave scattering, and hence arises in many applications. Therefore it is of practical importance for there to be robust numerical methods for the solution of the Helmholtz equation, which is the focus of this thesis.

Throughout this thesis we consider as our model problem the Helmholtz equation in a bounded domain subject to an impedance boundary condition which is an approximation of the so called *Sommerfeld radiation condition*. Once the PDE problem is then discretised with, say, low order finite elements the resulting system matrix is complex, symmetric and non-Hermitian. However for large values of the wavenumber k the solution of the Helmholtz equation is highly oscillatory and therefore a number of grid points growing at least as fast as $\mathcal{O}(k)$ must be chosen to ensure an accurate solution. The consequence of this is that for large k, or equivalently large domains, the system matrix will be very large. Therefore direct solvers are no longer a viable choice, and hence we turn instead to *cheaper* iterative solvers. In this thesis we consider the Schwarz domain decomposition as our choice of iterative method and preconditioner. Throughout we shall prove theoretical results which show that this algorithm converges with a number of iterations which grows like k with a fractional exponent. This exponent differs depending on the interface condition and overlap.

A new theory is developed for the optimised non-overlapping Schwarz method for the Helmholtz equation with an absorbing term. The optimised method works by first replacing the multiplier in the standard impedance condition with a general complex number determined by minimising the convergence rate of the Schwarz method. The

interface condition is known as an optimised interface condition. The consequence of this is that we can improve upon the previous methods with Dirichlet and standard impedance conditions. This is verified theoretically and then numerically by testing the Dirichlet, impedance and optimised interface conditions using the Schwarz method on some model problems.

There are however situations where we cannot solve the optimisation problem resulting from the Schwarz method exactly. This situation arises when we consider higher order approximations for the interface condition, and when we use overlap in the optimised method. Instead of solving the problem theoretically we solve it numerically to obtain the multiplier required for the optimised interface condition. What we observe in the numerical experiments is that we can improve the convergence of the Schwarz algorithm further.

Finally in this thesis we develop a new preconditioner which combines the ideas of the optimised Schwarz domain decomposition method with the sweeping precondtioner of Engquist and Ying to form a hybrid preconditioner. This hybrid preconditioner is used within GMRES, and tested in numerical computations on large scale geological problems, including some 3D examples. What we find is that this new preconditioner can converge very quickly. However, we also highlight areas which could be improved using other techniques which we outline as further work to be considered.

ACKNOWLEDGEMENTS

Firstly I would like to give my greatest thanks to my supervisor Ivan Graham at the University of Bath. In spite of a large workload, Ivan has always found time for meetings, to look over my work and give sound advice. I have been incredibly lucky to have you as a supervisor Ivan and appreciate all the support you have given to me throughout my PhD studies.

I would also like to extend a huge thank you to my industrial supervisor Paul Childs at Schlumberger Gould Research. I'm extremely grateful to Paul for our useful meetings in Cambridge and your feedback on each Chapter of my thesis.

A thank you is also extended to the excellent Numerical Analysis group in Bath who I was fortunate to be a part of. The weekly seminars were extremely enjoyable and helped me to gain a more broad education in mathematics, and provided vital feedback on my presentations. In particular I'd like to thank Euan Spence for helping me out on all things Helmholtz related, and Melina Freitag for helping me out on all things related to iterative methods. A thank you also goes to all of the administration staff in the Department of Mathematical Sciences and the computer support staff for the Faculty of Science. Also I would like to thank the EPSRC and Schlumberger for funding this project.

A special thank you also goes to all members (past and present) of the departmental football team for the countless trophies and memorable football trips around the UK and to Germany. In particular thanks to Acyr, Bati, Chris, Curdin, Elvijs, Fynn, Geoff, Huseyin, James C, James L, Matt, Miles, Owen Ray, Rachid and Tom.

Finally, I would like to thank my family for their support throughout all of my studies, and also to Fleur for putting up with me on a daily basis.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Motivation

The study of fast and robust numerical methods for computing the solution of wave propagation problems is an area of interest with a broad range of applications, such as medical physics and seismology. In the latter discipline a problem, which is of particular interest to those working in the oil and gas industry, is seismic imaging.

One particular example of seismic imaging as depicted in Figure 1-1 is described here. A boat goes out to sea trailing behind it a cable of hydrophones. A seismic shot (also referred to as a source) is fired from a source producing sound waves which propagate downwards, reflecting off the sea bed (and also sub-surface structures). These reflected waves (echoes) are then recorded by the hydrophones. (See the reference [16] for more details of the practical side of Seismology which is beyond the scope of this thesis.) Experimentally acquired reflective seismic data is then used to determine whether this particular location is good for drilling. The aim is to determine from the seismic data the characteristics of wave propagation in this particular area of sea, which tells us about the particular properties of the rock formations under the sea bed. From these rock properties we can determine if this domain of interest may contain hydrocarbons and is therefore a good location to drill for oil. This type of problem is an example of an *inverse* problem. One method to do the inversion is so called full waveform inversion (FWI) [55] . In this method we solve the inverse problem by solving the corresponding *forward* problem (in time-dependent form) and comparing that to our actual seismic data. The most commonly used forward model is the scalar *wave equation* (or its elastic

equivalent)

$$-\Delta U + \frac{1}{c^2}\frac{\partial^2 U}{\partial t^2} = F, \tag{1.1}$$

where Δ is the Laplacian, F is a forcing function (for example our source) and c is the speed of the waves at which the time and spatially varying wave U propagates. Then if we are given a starting model of the material properties in the domain we are interested in (for the above equation that would be the wave speed c) we then use the forward model to numerically simulate the propagation of acoustic waves in this domain given a certain source. The numerical results are then compared to the experimental seismic data, our starting model of the material properties is then updated using some suitable objective function and this process is carried on iteratively until the physical and simulated results reach a good level of agreement. Therefore at the end of this iterative process we should end up with a model which accurately describes the material properties in the domain of interest.

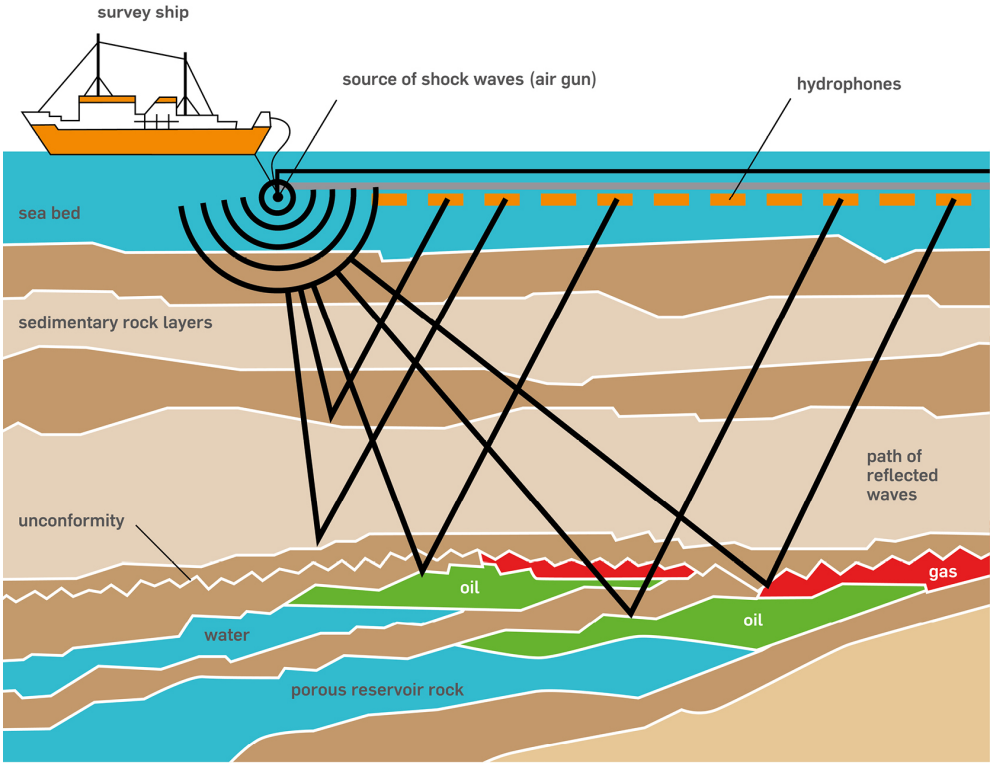

Figure 1-1: *Cartoon [43] of the typical scenario for the acquisition of reflective seismic data.*

The important thing to identify from the above process is that this inversion process involves solving the forward problem many times. Therefore it is of great interest to have numerical methods which solve the discrete form of (1.1) efficiently. Also we

should make the reader aware that the typical domain size for problems of this type are several kilometres in depth and often tens to hundreds of kilometres in width/length, and as the resulting waves oscillate many times the resulting linear systems which we solve become massive and pose problems for even the most efficient direct solvers. The reasons for this will be mentioned towards the end of this chapter.

1.2 The Helmholtz equation

Often it is more convenient to solve the forward problem (1.1) in the so called frequency domain [48], [49] as it results in solving a PDE problem which is time independent. We formulate the frequency domain form of the wave equation starting from (1.1). If we then consider that the waves $U(\mathbf{x}, t)$ are assumed to be time-harmonic (i.e. the time variation is sinusoidal) then we can separate the solution into those parts which are time and spatially varying in the following way

$$U(\mathbf{x}, t) := u(\mathbf{x})e^{-i\omega t},$$

where $\omega = 2\pi/\lambda$ is the angular frequency, and λ is the wavelength. It then follows that,

$$\frac{\partial^2 U}{\partial t^2} = -\omega^2 u(\mathbf{x})e^{-i\omega t}.$$

If we then substitute this into (1.1) this gives

$$-\Delta u(\mathbf{x})e^{-i\omega t} - \frac{\omega^2}{c^2}u(\mathbf{x})e^{-i\omega t} = F.$$

Therefore we can see that the wave equation reduces to the so called Helmholtz equation

$$\Delta u + k^2 u = -f. \tag{1.2}$$

where we assume $F(\mathbf{x}, t) = f(\mathbf{x})e^{-i\omega t}$. Here $k = \omega/c$ is known as the *wavenumber* and describes the spatial frequency of the wave. It is also possible by assuming that the waves are time harmonic to reduce the Maxwell equations and also the elastic wave equation to systems of Helmholtz type equations. Therefore the Helmholtz equation is integral to the study of wave propagation, and hence much research has gone into the study of its solutions and the efficient numerical computation of these solutions. We will now formulate the model exterior scattering problem and show how this relates to the interior impedance problem which we consider as the model problem for our

study.

1.3 The exterior scattering problem and the model interior problem

We consider the model Helmholtz problem on an unbounded domain in the context of an exterior scattering problem. We define this problem as the following.

Definition 1.1:
The exterior Dirichlet problem for the Helmholtz equation is as follows. We seek a solution u which satisfies the Helmholtz equation in the exterior domain, $\Omega' := \mathbb{R}^d \setminus \overline{\Omega}$ (where d denotes the dimension), to the scatterer, Ω, which has a Lipschitz boundary Γ. That is we solve

$$\left.\begin{aligned} \Delta u + k^2 u = 0, \ \textit{in } \Omega', \\ u = -u^I, \ \textit{on } \Gamma, \end{aligned}\right\} \tag{1.3}$$

where u_I is the incident wave. Furthermore u satisfies the Sommerfeld radiation condition,

$$\frac{\partial u}{\partial r} - iku = o\left(r^{\frac{1-d}{2}}\right), \ \textit{as } r \to \infty. \tag{1.4}$$

Physically there is an incident wavefield $u_I(\mathbf{x})$ which propagates in Ω', and interacts with the boundary Γ of an obstacle Ω producing a scattered field u. This scattered wavefield satisfies (1.3) and the far field condition (1.4) which ensures that all scattered waves are absorbed at infinity.

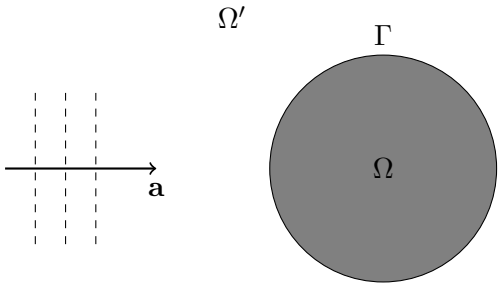

Figure 1-2: *Illustration of scattering in $2D$. In this plot an incident wavefield u^I is propagating in the direction of the vector $\mathbf{a}$. Our scatterer is denoted Ω with it's boundary Γ.*

This problem is known to have a unique solution for all k [12]. Although there are techniques for tackling (1.3), (1.4) on unbounded domains, most numerical methods for solving the Helmholtz equation consider approximating the problem on a bounded

domain. If one truncates the domain of the previously considered exterior scattering problem to a ball of radius R, say B_R (see Figure 1-3), then provided R is large enough [29] we can place an absorbing boundary condition on B_R which approximates the radiation condition (1.4). The simplest choice is the *impedance* boundary condition (or sometimes called Robin condition) of the form,

$$\frac{\partial u}{\partial n} - iku = 0, \text{ on } B_R, \tag{1.5}$$

where $\partial/\partial n$ denotes the outgoing normal derivative from B_R. Thus the approximation

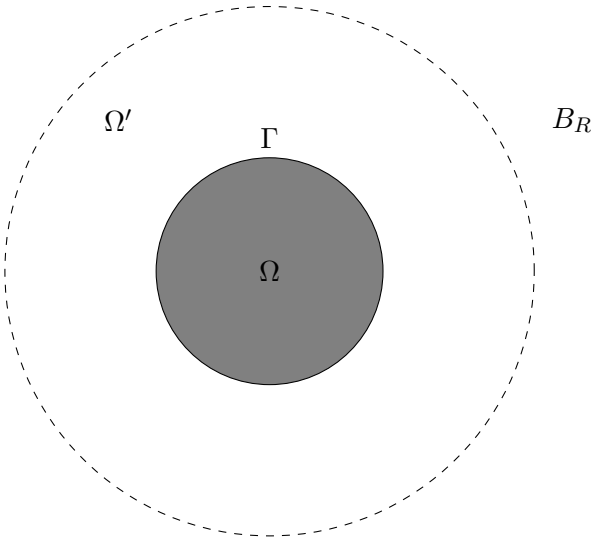

Figure 1-3: *Illustration of the truncation of the previous exterior scattering problem of Figure 1-2 to a ball of radius R, B_R.*

of (1.3), (1.4) consists of the following problem on Ω'

$$\left.\begin{aligned} \Delta u + k^2 u = 0, \text{ in } \Omega', \\ u = -u^I, \text{ on } \Gamma, \\ \frac{\partial u}{\partial r} - iku = 0, \text{ on } B_R. \end{aligned}\right\} \tag{1.6}$$

An even simpler model problem is the one obtained by ignoring the scatterer Ω altogether. This is the so called interior impedance problem where we are solving for the wavefield inside a bounded domain Ω with an impedance boundary condition on $\partial\Omega$. This is stated in (1.7). We include a forcing term f on the right hand side of the Helmholtz equation. In this case the incident field would be generated from an internal (for example a point source) or an external force (for example an incoming plane wave). We consider also that the resulting scattered field is created from reflections at the boundary of the domain or possibly internally if the wavenumber k

varies throughout the domain, and hence for simplicity we omit the scatterer Ω' in the following definition.

Definition 1.2:
We say that u satisfies the interior impedance problem

$$\left.\begin{aligned} &\Delta u + k^2 u = -f, \ \text{in } \Omega \subset \mathbb{R}^d, \\ &\frac{\partial u}{\partial n} - iku = g, \ \text{on } \partial\Omega, \end{aligned}\right\} \tag{1.7}$$

where Ω is a bounded Lipschitz domain in $\mathbb{R}^d$ with boundary $\partial\Omega$, and f and g are functions to be chosen.

Most of this thesis is about the solution of (1.7). This is used as a model problem in many other studies [38], [39], [44], [59] (among many others). The boundary value problem (1.7) also has a unique solution for all k [37]. If for example $f \in L^2(\Omega)$ and $g \in L^2(\partial\Omega)$ then there is a unique $u \in H^1(\Omega)$ which satisfies (1.7) in weak form. Analytical solutions are very difficult to obtain or no longer possible for most interesting applications, for example if the wavenumber k is highly variable in the domain Ω. Therefore we consider discretising (1.7) using the finite element method (FEM) and computing the solution numerically. We could use other discretisation schemes and in Chapter 5 we discuss the corresponding discretisation using the finite difference method.

1.4 The finite element method (FEM)

We start by disctretising (1.7) by the conventional finite element method and forming the resulting linear system. Afterwards we discuss some of the issues that the conventional finite element method has when approximating solutions of the Helmholtz equation as k becomes large.

We start by writing (1.7) in so called *weak* variational form, that is we multiply the first equation of (1.7) by a test function $v \in H^1(\Omega)$ and integrate over Ω,

$$\int_\Omega \Delta u \bar{v} + k^2 \int_\Omega u\bar{v} = -\int_\Omega f\bar{v}.$$

We then use Green's first identity (or one can think of this as integrating by parts in the given spatial dimension) on the left hand side of the above equation to give,

$$-\int_\Omega \nabla u \nabla \bar{v} + \int_{\partial\Omega} \bar{v}\frac{\partial u}{\partial n} + k^2 \int_\Omega u\bar{v} = -\int_\Omega f\bar{v}.$$

We can then use the boundary equation of (1.7) to treat the $\partial u/\partial n$ term,

$$-\int_\Omega \nabla u \nabla \bar{v} + \int_{\partial\Omega} \bar{v}\left(g + iku\right) + k^2 \int_\Omega u\bar{v} = -\int_\Omega f\bar{v}.$$

Rearranging this then gives the following *weak* form for the interior impedance problem (1.7)

$$\int_\Omega \nabla u \nabla \bar{v} - k^2 \int_\Omega u\bar{v} - ik \int_{\partial\Omega} u\bar{v} = \int_\Omega f\bar{v} + \int_{\partial\Omega} g\bar{v}.$$

Therefore our problem now is to find $u \in H^1(\Omega)$ such that

$$a(u,v) = b(v), \text{ for all } v \in H^1(\Omega), \tag{1.8}$$

where

$$a(u,v) = \int_\Omega \nabla u \nabla \bar{v} - k^2 \int_\Omega u\bar{v} - ik \int_{\partial\Omega} u\bar{v}, \text{ and,} \tag{1.9}$$

$$b(v) = \int_\Omega f\bar{v} + \int_{\partial\Omega} g\bar{v}. \tag{1.10}$$

The next step in the finite element method is to decompose our bounded domain Ω into a mesh of elements e.g. triangles (2D) or tetrahedra in (3D), where we define the mesh width as h. An example of this is given in Figure 1-4. The vertices of these elements we call the nodes n_i of the mesh, where there are N nodes in total and hence $i = 1, ..., N$. We now choose to use the simplest choice of piecewise linear finite elements (hat functions) on our mesh. To start we define the set of basis functions $\phi_i : \Omega \to \mathbb{R}$ which are defined at the nodes n_i in the following way

- $\phi_i(n_i) = \delta_{ij}$,
- ϕ_i is linear on each element of the mesh, and
- ϕ_i vanishes on elements which do not contain n_i.

If we return to the problem (1.8), we approximate the problem by seeking $u_h \in V_h := \text{span}\{\phi_i\}$ such that

$$a(u_h, v_h) = b(v_h), \text{ for all } v \in V_h. \tag{1.11}$$

by parts Writing

$$u_h = \sum_{j=1}^{N} U_j \phi_j, \tag{1.12}$$

and noting that $a(.,.)$ is a sesquilinear form, we see that (1.11) is equivalent to

$$\sum_{j=1}^{N} a\left(\phi_j, \phi_i\right) U_j = b\left(\phi_i\right), \ i = 1, \ldots, N.$$

by parts If we recall the form of (1.9) and (1.10) then the above equation can be written as the following linear system of equations,

$$A\mathbf{U} = \mathbf{b}, \tag{1.13}$$

where we solve for the unknown coefficients $\mathbf{U} \in \mathbb{C}^N$ and substitute these into our ansatz (1.12) to obtain the solution of (1.8). The majority of this thesis is about the

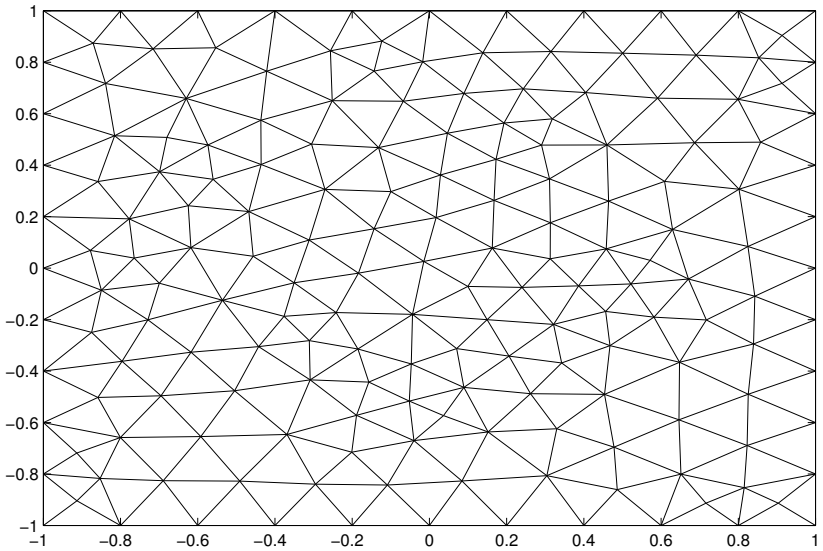

Figure 1-4: *Example of a finite element mesh on* $\Omega = (-1, 1)^2$

efficient iterative solution of (1.13)

Before we discuss solving (1.13) we first recall the difficulties in using standard low order methods such as piecewise linear finite elements to discretise the Helmholtz probelm (1.7) for large k. Firstly as solutions of the Helmholtz equation are oscillatory and become more so when k increases (the oscillations occur on a scale of $1/k$ in general) one cannot keep a fixed number of grid points. Therefore the convention has been to take 10 grid points per wavelength to try to maintain accuracy. This means that N is proportional to k^d and hk is bounded. However even this is not enough in some cases

when k gets very large due to the so called *pollution effect*. For example in [28, §3.4] the authors consider the following 1D model Helmholtz problem

$$\begin{aligned} \partial_x^2 u(x) + k^2 u(x) &= f(x), \text{ for } x \in (0,1), \\ u(0) &= 0, \\ \partial_x u(1) - iku(1) &= 0, \end{aligned}$$

The authors show that the relative error of the finite element method in the H^1 norm has the following bound (assuming that hk is constant)

$$\frac{\|u - u_h\|_{H^1}}{\|u\|_{H^1}} \leq C_1 hk + C_2 k^3 h^2,$$

where u_h is the finite element approximate solution and C_1, C_2 are constants independent of k. Therefore the above bound tells one that even if hk is chosen small enough so as to control the first term (the approximation error) on the right hand side, the second term (the effect due to numerical pollution) cannot be controlled and the relative error may blow up. To bound the relative error independently of k then a more restrictive assumption that $hk^{\frac{3}{2}}$ is bounded should be adopted. In higher dimensions it is conjectured that having $hk^{\frac{3}{2}}$ bounded is also sufficient to remove pollution but such a result is not known rigorously and only hk^2 bounded is known to be sufficient. The consequence of this is that the growth of the total number of grid points $N \sim k^{2d}$ which leads to very large system matrices A which can make the method numerically intractable for very large k.

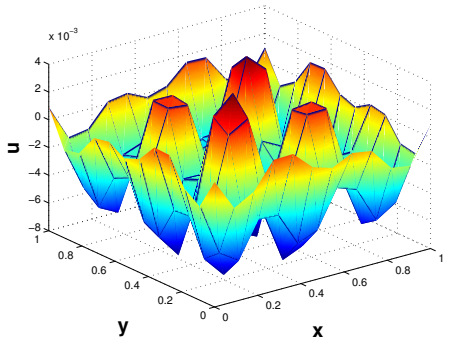

Figure 1-5: *Solution of* (1.13) *where* $\boldsymbol{b} = 1$, $k = 20$ *and* $hk = \frac{5}{3}$. *A has 144 nodes.*

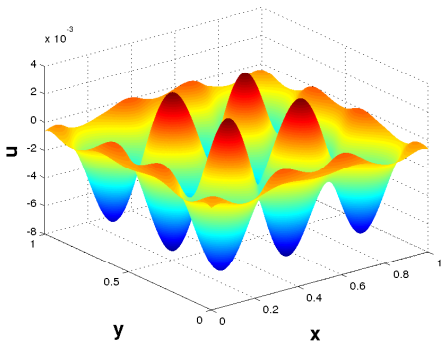

Figure 1-6: *Solution of* (1.13) *where* $\boldsymbol{b} = 1$, $k = 20$ *and* $hk^2 < 1$. *A has 1048576 nodes.*

Much research [11], [25], [37] (among others) has been dedicated to numerical discretisation methods which reduce the effects of numerical pollution for variational formulations of the Helmholtz equation . For example the ultra weak variational formulation

(UWVF) [11] which uses fundamental solutions of the Helmholtz equation as its basis functions, or other methods [41] which formulate a variational form which is coercive. However these methods are somewhat orthogonal to the ideas in this thesis, and also they are more complicated to implement due to the choice of more exotic basis functions.

The pollution effect is mostly mentioned for large k on domains of unitary size which are often referred to as high frequency problems as $k = \omega/c$ where ω is the angular frequency. Consider instead that our domain is very large, say $\mathbf{x} \in (0, L)^2$ where $L \gg 1$, and our Helmholtz equation is given by

$$\Delta_{\mathbf{x}} u(\mathbf{x}) + \left(\frac{\omega}{c}\right)^2 u(\mathbf{x}) = 0, \quad \mathbf{x} \in (0, L)^2$$

However we could rescale the spatial coordinates to relate the large domain and one on a unit sized domain. If we replace $\mathbf{x}$ by $\hat{\mathbf{x}} = \frac{\mathbf{x}}{L}$, and hence $\Delta_{\mathbf{x}}$ is replaced by $\frac{1}{L^2}\Delta u_{\hat{\mathbf{x}}}$, then our rescaled Helmholtz equation becomes

$$\frac{1}{L^2}\Delta u_{\hat{\mathbf{x}}}(\hat{\mathbf{x}}) + \left(\frac{\omega}{c}\right)^2 u(\hat{\mathbf{x}}) = 0, \quad \hat{\mathbf{x}} \in (0, 1)^2.$$

Therefore

$$\Delta u + \left(\frac{\omega L}{c}\right)^2 u = 0.$$

So we see that problems on very large domains $L \gg 1$ with moderate k are equivalent to high frequency problems on a domain of unitary size. Therefore accurate discretisation schemes are also of importance to those working on large industrial scale problems with moderate values of k.

1.5 Solution of the linear system

We now return to our linear system of equations (1.13) where the system matrix takes the form,

$$\begin{aligned} A_{i,j} &= \int_\Omega \nabla\phi_i \nabla\phi_j - k^2 \int_\Omega \phi_i\phi_j - ik \int_{\partial\Omega} \phi_i\phi_j, \\ &= S_{i,j} - k^2 M_{i,j} - ikB_{i,j} \end{aligned} \tag{1.14}$$

and the load vector take the form

$$b_j = \int_\Omega f\phi_j + \int_{\partial\Omega} g\phi_j,$$

where ϕ_j are our piecewise linear basis functions and $i, j = 1, \ldots N$. The matrices S, M and B are referred to respectively as the stiffness matrix, mass matrix and boundary mass matrix. These matrices are sparse, symmetric and have real valued entries and therefore the system matrix A will be sparse, symmetric but has complex entries and is non-Hermitian. When N is small enough one would just solve (1.13) using a direct solver but as mentioned previously our system matrix can be very large (for some 3D models $> \mathcal{O}\left(10^8\right)$) and therefore the memory required to form the matrix may be prohibitive. Therefore we could consider using an iterative method to solve the linear system as many iterative methods require only the action of A multiplied by a vector and so the large matrix A doesn't need to be formed. However using an iterative method alone is not an option as the matrix A is not well enough behaved and there are very few convergence results for solving it iteratively. Indeed in general one expects that iterative methods without preconditioning will perform very badly when solving (1.13).

This thesis is concerned with the construction of effective preconditioners P to allow for fast and accurate numerical solutions of the preconditioned linear system,

$$PA\mathbf{U} = P\mathbf{b} \tag{1.15}$$

using iterative solvers such as GMRES [57]. We shall introduce this particular iterative solver and the preconditioning techniques considered in the next chapter. We write (1.15) in the left preconditioning form but most of what we do can also be used in right preconditioning.

The class of preconditioner that we consider are the so called *shifted* Laplace type preconditioners which are formed from the following problem

$$\left.\begin{aligned} \Delta u + k^2 u + i\epsilon u = -f, \text{ in } \Omega, \\ \frac{\partial}{\partial n} u - i\eta u = g, \text{ on } \partial\Omega, \end{aligned}\right\} \tag{1.16}$$

where $\eta, \epsilon \in \mathbb{R}_+$ and all other variables are defined as previously for (1.7). If the boundary value problem (1.16) is discretised with piecewise linear finite elements then we obtain the following linear system,

$$A_\epsilon \mathbf{U} = \mathbf{b}. \tag{1.17}$$

The advantage of using (1.16) is that the introduction of the term $i\epsilon$ introduces artificial damping into the problem. The consequence of this is that the resulting system matrix A_ϵ is *better behaved* than A, in the sense that the corresponding boundary of the field of values of A_ϵ is bounded away from the origin (this will be discussed in more detail

in the next chapter). Therefore solving (1.17) using a standard iterative method is no longer as difficult. We then form preconditioners P by approximating the inverse of A_ϵ. We shall define these preconditioners as B_ϵ^{-1}.

We choose to use domain decomposition to approximate the inverse of A_ϵ and hence construct P in (1.15), one could use another method such as one Multigrid v-cycle as was used in [58], [59]. In Chapter 5 we recall the sweeping preconditioner [5]. Since this method requires the direct solution of many large subproblems (especially in 3D) we consider modifications in which the direct solve is replaced by a domain decomposition approximation. More details of these other methods is discussed in §2.2 and in Chapter 5. In §2.2 it is shown that if we choose ϵ such that:

(1) our approximate inverse B_ϵ^{-1} is an effective preconditioner for the system (1.17) and

(2) A_ϵ^{-1} (the exact inverse) is an effective preconditioner of (1.13)

then B_ϵ^{-1} is also an effective preconditioner for (1.13). However the requirements posed by (1) and (2) are contradictory. In order for (1) to be true then ideally ϵ should be large, as when ϵ gets larger then the solving the system (1.17) by an iterative method becomes easier (this is explained further in Chapter 3). However for (2) to be satisfied ϵ should be as small as possible. The reason for this is that when $\epsilon = 0$ then $A^{-1}A = \mathbf{1}$ which is the optimal scenario for an iterative solver such as GMRES. Therefore choosing an ϵ which satisfies both (1) and (2) is not a trivial task, and a rigorous analysis for this remains an open question. To display this we show in Figures 1-7 and 1-8 the numerical solutions of (1.13) (left) and (1.17) (right, with $\epsilon = k^2$) respectively with $k = 60$, $\mathbf{b} = \mathbf{1}$ and $hk^{\frac{3}{2}}$ bounded. It is visible that introducing an artificial damping greatly reduces the number of oscillations in the solution and iterative solution. However a consequence of introducing too much damping is that solutions of (1.17) do not closely resemble those of original problem (1.13) that we want to solve. Therefore one would naively not expect $A_\epsilon^{-1}A$ to be close to 1.

In chapters 2, 3 and 4 we develop methods which solve (1.17) quickly and discuss the construction of good preconditioners B_ϵ^{-1} for (1.17). Then in Chapter 5 we use these methods to motivate the choice of P in (1.15).

We now discuss the contributions of this thesis and then outline the contents of this thesis.

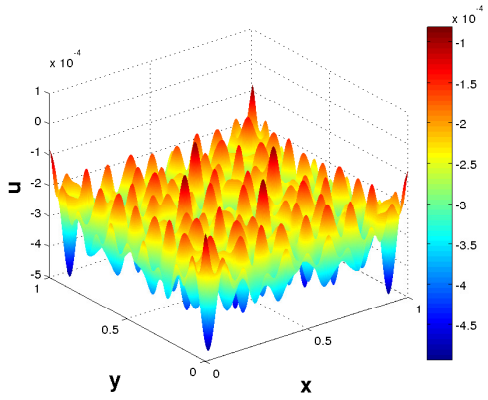

Figure 1-7: *Solution of* (1.13) *where* $\boldsymbol{b} = 1$, $k = 60$ *and* $hk^{\frac{3}{2}} = 1$. *A has 872356 nodes.*

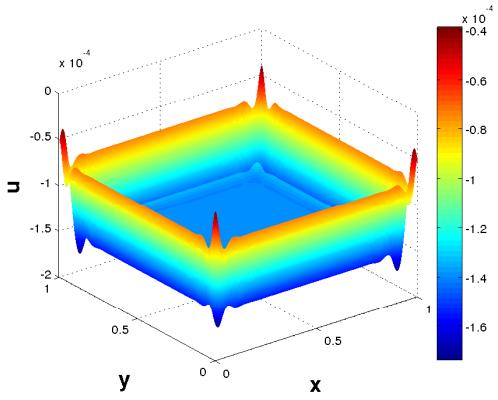

Figure 1-8: *Solution of* (1.17) *where* $\boldsymbol{b} = 1$, $k = 60$, $\epsilon = k^2$ *and* $hk^{\frac{3}{2}} = 1$. A_ϵ *has 872356 nodes.*

1.6 Main contributions of this thesis

This thesis contributes the following:

1. The alternating Schwarz algorithm [52] is applied to the interior Helmholtz problem with absorbing term $i\epsilon$ (1.16). In a model two domain setting a Fourier analysis is performed on this algorithm to derive its rate of convergence ρ which can be found in Theorem 2.10. Then asymptotic expressions are calculated for the maximum of the convergence rate as $k \to \infty$ when a Dirichlet (Corollary 2.23) or impedance condition (Theorem's 2.27, 2.28, 2.29 and 2.30) is used on the interface between subdomains. It is shown that when the Dirichlet condition is used overlap between subdomains is required for the algorithm to converge, see Theorem 2.21. However, when an impedance condition is used the non-overlapping algorithm converges.

2. Following from this, in Chapter 3, a zeroth order optimised interface condition is calculated by solving a minimax problem involving the convergence rate for the non-overlapping Schwarz method. The result is given in Theorem 3.2. The maximum of the convergence rate is then computed, see Corollary 3.3, for increasing k with this new choice of interface condition and it can be seen that the rate of convergence now does not degrade as strongly as $k \to \infty$, compared to the the case when Dirichlet or impedance condition is used. Furthermore when $\epsilon = k^2$ it is shown that the number of Schwarz iterations is bounded independently of k, for k increasing.

3. Zeroth order optimised conditions for the overlapping Schwarz method and second order optimised for the non-overlapping Schwarz method are studied numerically

in Chapter 4. We show, in Corollaries 4.2 and 4.5, that compared to the performance of the non-overlapping Schwarz method, if we either use overlap or use a higher order interface condition we reduce the growth with respect to k of the maximum of the convergence rate of the Schwarz algorithm. We observe a further improvement in performance in our numerical computations

4. Finally in Chapter 5 a new Hybrid preconditioner (P in (1.15)) is introduced which replaces the direct solves in the sweeping preconditioner [5] with GMRES preconditioned with an optimised Schwarz method where the interface conditions on subdomains use the optimised Schwarz interface conditions calculated in this Thesis. The numerical method is then tested by solving (1.15) using some large scale industrial model data including problems in 3D. The number of iterations taken by the new algorithm are shown to scale well when k increases.

1.7 Outline of this thesis

The structure of the thesis is the following:

- In Chapter 2 we give a detailed introduction to the generalized minimal residual method (GMRES) and review some convergence theory results involving the field of values of the system matrix. We then introduce the so called *shifted* Laplace preconditioner. A short literature review §2.2.1 of shifted Laplace preconditioners is given where we discuss the works of previous authors and their findings.

  The Schwarz method is then introduced in section 2.3 and a Fourier analysis of the two subdomain algorithm for the iterative solution of (1.16) is given. This results in the calculation of an expression for the convergence rate of the algorithm. In the remainder of the chapter we then examine how the choice of interface condition and the use of overlap influence the maximum of the convergence rate. We do this by looking at the behaviour of the leading order terms of the maximum of the convergence rate for increasing k. Considerable analysis of asymptotics is required.

- In Chapter 3 we develop a zeroth order optimised interface condition for the non-overlapping Schwarz method. The parameter in this interface condition is found by solving a minimax problem involving the maximum of the convergence rate. The analysis is rigorous and novel. We then show that the maximum of the convergence rate of the non-overlapping Schwarz method with zeroth order interface condition does not grow as quickly with k compared to the Schwarz method with interface conditions examined in Chapter 2 as k increases.

- In Chapter 4 we present two numerical studies involving the Schwarz method. In the first set we examine a zeroth order optimised interface condition for the overlapping Schwarz method and also a second order optimised interface condition for the non-overlapping Schwarz method. The parameters used for both of these interface conditions are found by numerically solving a given minimax problem. The results of these numerical experiments are then used to conjecture how the parameters in the optimised interface conditions grow with k for k increasing. We also examine the influence these new interface conditions have on the growth of the maximum of the convergence rate for k increasing. It is shown that both of these new conditions lead to a better iterative method than that studied in chapters 2, 3.

  In the second set of numerical experiments we implement the two subdomain Schwarz method as an iterative solver for the solution of (1.17). We then test the resulting convergence of some of the methods of Chapter 2, 3 and 4 in terms of the number of iterations taken to reach a user defined exit criterion. What we show is that using optimised methods do indeed improve the convergence of the Schwarz method compared to the standard impedance condition. Also that using even the smallest amount of overlap gives a noticeable improvement in convergence. We also test the Schwarz method as a preconditioner for the system (1.17) and observe similar results.

- In Chapter 5 we start by giving an introduction to the sweeping preconditioner [5] by relating it to the classical Thomas algorithm for solving tri-diagonal systems. We then introduce a new hybrid preconditioner which replaces the direct solves used in the sweeping preconditioner with GMRES preconditioned with an optimised Schwarz domain decomposition method. Details of the method are given and finally numerical results are performed on large scale 2D and 3D industrial model problems. The numerical results show that this method shows a small growth in the number of iterations as k increases, however some work needs to be done to improve its scalability in 3D.

- Lastly in Chapter 6 we provide a brief review of the thesis and some concluding remarks, along with some comments on future works and extensions on what has been done in this thesis.

CHAPTER 2

ITERATIVE METHODS AND PRECONDITIONING

In this chapter we start by briefly introducing the Generalised Minimal Residual method (abbreviated as GMRES) an iterative solution method for solving linear systems of equations, developed by Saad and Schultz [57]. The complexity of GMRES is then be considered, before moving on to a discussion on preconditioning techniques which aim to speed up its convergence. The Field of Values of a matrix will be defined and its relation to the convergence of GMRES discussed.

After this general introduction a review of the more specific use of absorption in the construction of preconditioners for the Helmholtz equation will be given and convergence results presented.

Finally we will introduce domain decomposition methods and we will explain how they can function either as iterative methods or as preconditioners. Firstly we will discuss a more general framework for domain decomposition and then the specific optimised variant that we focus on.

For clarity we redefine the interior model Helmholtz problem (defined previously in (1.7)) that we consider solving,

$$\left.\begin{aligned} \Delta u + k^2 u &= -f, \text{ in } \Omega \subset \mathbb{R}^d \text{ where } d = 2, 3, \\ \frac{\partial}{\partial n} u - iku &= g, \text{ on } \partial\Omega, \end{aligned}\right\} \tag{2.1}$$

where $k \in \mathbb{R}_+$, $\frac{\partial}{\partial n}$ denotes the normal derivative and Ω is a bounded domain. We also specify that $f \in \mathcal{L}^2(\Omega)$ and $g \in \mathcal{L}^2(\partial\Omega)$. We recall from § 1.4 that discretising (2.1)

using low-order finite elements results in a linear system of the form,

$$A\mathbf{U} = \mathbf{b}, \tag{2.2}$$

where we recall from (1.14) that,

$$A = S - k^2 M - ikB,$$

Hence A is a complex, non-Hermitian (as $A \neq A^*$), large and sparse matrix. The matrices S, M and B are given in (1.14). In the following we consider solving (2.2) with an iterative method.

2.1 Krylov methods and GMRES

Iterative methods are commonly used when the system matrix A is difficult to invert, it could be dense or very large, the latter being the case with applications of the Helmholtz equation. Iterative methods are in a way very intuitive: one starts with an initial guess $\mathbf{U}_0$ and then generates consecutive solutions $\mathbf{U}_m$, by some iterative process. Which are, one hopes, more accurate approximations to the solution of (2.2). GMRES falls into a category of iterative methods known as *Krylov methods*. Such methods are characterised by choosing each successive approximate solution, $\mathbf{U}_m$, from a Krylov space $\mathcal{K}_m(A, b)$, which is defined as follows.

Definition 2.1:
The Krylov subspace of order m is the linear space generated by the $n \times n$ matrix A and vector $\boldsymbol{b} \in \mathbb{R}^n$, defined as,

$$\mathcal{K}_m(A, \boldsymbol{b}) := span\{\boldsymbol{b}, A\boldsymbol{b}, A^2\boldsymbol{b}, \ldots, A^{m-1}\boldsymbol{b}\}. \tag{2.3}$$

GMRES chooses its next approximate solution by finding the $\mathbf{U}_m \in \mathcal{K}_m(A, b)$ which minimises the Euclidean norm of the residual, $\mathbf{b} - A\mathbf{U}_m$. However the vectors $\{\mathbf{b}, A\mathbf{b}, \ldots, A^{m-1}\mathbf{b}\}$, which serve as a basis for (2.3), can become nearly linearly dependent. Consequently it is necessary to use some orthogonalisation method to construct a set of vectors $\mathbf{q}_1, \ldots, \mathbf{q}_m$ which are also a basis for (2.3). This is achieved using Arnoldi's method which uses a modified Gram-Schmidt process to construct an orthonormal basis for (2.3). Modified Gram-Schmidt is used as it ensures orthogonality of each newly constructed vector, even if there are pre-existing rounding errors. This is not always true of classical Gram-Schmidt [36] which can suffer from a loss of orthogonality and thus instability in the method.

The Arnoldi method starts by using $\mathbf{q}_1 = \frac{\mathbf{b}}{\|\mathbf{b}\|_2}$ as the basis for $\mathcal{K}_1(A, \mathbf{b})$. The next basis vector, $\mathbf{q}_2$ for $\mathcal{K}_2(A, \mathbf{b})$, is formed by orthogonalising the vector $A\mathbf{q}_1$ such that $\mathbf{q}_2 = A\mathbf{q}_1 - (A\mathbf{q}_1, \mathbf{q}_1)$, where $(.,.)$ denotes an inner product. This process is then repeated iteratively to form the orthonormal basis $\{\mathbf{q}_1, \ldots, \mathbf{q}_m\}$ for (2.3), and the vectors $\mathbf{q}_i$ $(i = 1 \ldots m)$ are known as Arnoldi vectors. A pseudocode for the Arnoldi method is presented in Algorithm 1. If we define $Q_m = (\mathbf{q}_1 \quad \ldots \quad \mathbf{q}_m)$ to be the $n \times m$

Algorithm 1 Arnoldi iteration

```
Define q_1 = b / ||b||_2
for j = 1, 2, ..., m do
    q̂_{j+1} = A q_j
    for i = 1, 2, ..., j do
        h_{i,j} = (q̂_{j+1}, q_i)
        q̂_{j+1} = q̂_{j+1} − h_{i,j} q_i
    end for
    h_{j+1,j} = ||q̂_{j+1}||_2
    q_{j+1} = q̂_{j+1} / h_{j+1,j}
end for
```

matrix consisting of the Arnoldi vectors as its columns then it is possible to show that the Arnoldi method leads to the following matrix decomposition,

$$AQ_m = Q_{m+1}H_{m+1,m}. \tag{2.4}$$

In the above equation $H_{m+1,m}$ denotes an upper Hessenberg matrix of size $m+1 \times m$. (An Upper Hessenberg matrix is an upper triangular matrix which may have additional non-zero elements on the off diagonal below the diagonal.)

As we stated previously, the second stage of the GMRES algorithm is concerned with solving a least squares problem given by (2.5) below. Therefore, using what we know about the Arnoldi method, if we start the GMRES algorithm with an initial guess $\mathbf{U}_0$ then the mth iterate is given by,

$$\mathbf{U}_m = \mathbf{U}_0 + Q_m\mathbf{z}_m,$$

where $\mathbf{z}_m$ is the solution of the following least squares problem,

$$\min_{\mathbf{z}_m \in \mathbb{C}_m} \|\mathbf{b} - AQ_m\mathbf{z}_m\|_2, \tag{2.5}$$

i.e. $\mathbf{z}_m$ is chosen such that it minimises the Euclidean norm of the residual. Combining (2.4) and (2.5) results in,

$$\min_{\mathbf{z}_m \in \mathbb{C}_m} \|\mathbf{b} - AQ_m\mathbf{z}_m\|_2 = \min_{\mathbf{z}_m \in 2015-04-29-14:04\mathbb{C}_m} \|\mathbf{b} - Q_{m+1}H_{m+1,m}\mathbf{z}_m\|_2. \tag{2.6}$$

Noting that $\mathbf{b} = \mathbf{q}_1\|\mathbf{b}\|_2 = Q_{m+1}\mathbf{e}_1\|\mathbf{b}\|_2$, where $\mathbf{e}_1 = (1, 0, \ldots, 0)^T$ of size $m+1$, (2.6) becomes,

$$\begin{aligned}\min_{\mathbf{z}_m \in \mathbb{C}_m} \|\mathbf{b} - Q_{m+1}H_{m+1,m}\mathbf{z}_m\|_2 &= \min_{\mathbf{z}_m \in \mathbb{C}_m} \left\| Q_{m+1}\mathbf{e}_1\|\mathbf{b}\| - Q_{m+1}H_{m+1,m}\mathbf{z}_m \right\|_2, \\ &= \min_{\mathbf{z}_m \in \mathbb{C}_m} \left\| \mathbf{e}_1\|\mathbf{b}\|_2 - H_{m+1,m}\mathbf{z}_m \right\|_2\end{aligned}$$

Therefore the least squares problem that we have to solve at iteration m of GMRES has been simplified to the following,

$$\min_{\mathbf{z}_m \in \mathbb{C}_m} \left\| \mathbf{e}_1\|\mathbf{b}\|_2 - H_{m+1,m}\mathbf{z}_m \right\|_2 \tag{2.7}$$

The standard approach to solve (2.7) is to use QR factorisation, for details of this we refer the reader to the following reference [56]. We can now present a pseudocode for GMRES in Algorithm 2.

Algorithm 2 GMRES algorithm

Given an exit tolerance of τ.
Initialise $\mathbf{U}_0 = 0$.
Set $\mathbf{q}_1 = \frac{\mathbf{b}}{\|\mathbf{b}\|_2}$, $Q_1 = \mathbf{q}_1$.
for $i = 1, 2, \ldots,$ **do**
    Compute $\mathbf{q}_{i+1}$ using Arnoldi's method.
    Update $Q_{i+1} = \begin{pmatrix} Q_i & \mathbf{q}_{i+1} \end{pmatrix}$.
    Solve (2.7) using QR method to find $\mathbf{z}_i$.
    Update solution $\mathbf{U}_i = \mathbf{U}_0 + Q_{i+1}\mathbf{z}_i$.
    if $\frac{\|A\mathbf{U}_i - \mathbf{b}\|_2}{\|\mathbf{b}\|_2} < \tau$ **then**
        Exit
    end if
end for

Note in the above algorithm that we have included an exit criterion. In practice one does not want to run the GMRES algorithm to completion, that is when the exact solution is achieved. Rather one stops when an iterate is deemed to be close enough to the solution of (2.2). Here we use the criterion that the relative residual $\frac{\|A\mathbf{U}_i - \mathbf{b}\|_2}{\|\mathbf{b}\|_2}$ be less than given a tolerance τ, which is chosen to be small (say 10^{-6}).

We recall that the GMRES algorithm has two main stages: firstly the computation of the Arnoldi basis vectors by the Arnoldi algorithm, and then the solution of the linear least squares problem and the update of the solution. We can see from Algorithm 1 that the dominating computational cost of GMRES appears to be the matrix vector product $A\mathbf{q}_j$, which for a general dense matrix would cost $\mathcal{O}(n^2)$. However, as A is a sparse matrix which reduces the cost to $O(n)$. The other overhead is the Gram-Schmidt process itself which costs $O(m^2 n)$ [23, §5.2.8].

The remaining cost in the GMRES algorithm comes from solution of the least squares problem (2.7). This can at best be $O(m^2 n)$ using an efficient QR algorithm as mentioned in [33, § 3.5]. Hence the overall computational complexity of the GMRES algorithm is $O(m^2 n)$. For fixed n the complexity grows as $\mathcal{O}(m^2)$, where m is the iteration number, if the number of iterations becomes quite large (which is often the case without the application of a preconditioner) then the algorithm can become costly. A variant of GMRES which can help is restarted GMRES, commonly called GMRES(r). In GMRES(r) the full GMRES algorithm is restarted every r iterations. The current iterate is then used as the new initial guess and restarts the iteration. This can improve the rate of convergence of GMRES, and cut down on the storage and amount of work to be done. However there is little theoretical guidance as to how to best choose the restart parameter.

2.2 Convergence of GMRES and Preconditioning the Helmholtz equation

We now briefly introduce some convergence results for GMRES. Much of the literature on convergence of Krylov methods, see [24] for example, deals with the case when the system matrix is Hermitian and involves condition number estimates. This is not useful for the solution of (2.2) as the system matrix is non-Hermitian. We present convergence results from [2] and [17], which give an upper bound for the relative residual for GMRES involving the field of values of the system matrix. Finally we introduce some recent work which examines how (at least in theory) one can precondition the specific system (2.2) arising from (2.1) such that GMRES converges independently of the wavenumber k.

We start by defining the field of values for a matrix, and some general convergence results for GMRES.

Definition 2.2:
The field of values (or numerical range) of the $n \times n$ matrix B is the following,

$$W(B) := \{(B\boldsymbol{x}, \boldsymbol{x}) : \boldsymbol{x} \in \mathbb{C}^n, \|\boldsymbol{x}\|_2 = 1\}, \tag{2.8}$$

where $(.,.)$ denotes the Euclidean inner product.

The field of values contains all of the eigenvalues. It is easy to see this as if $\mathbf{v}$ is a unit eigenvector of a matrix B with corresponding eigenvalue λ, that is $B\mathbf{v} = \lambda\mathbf{v}$ where $\|\mathbf{v}\|_2 = 1$. Then it follows that $\lambda = \mathbf{v}^* B\mathbf{v} \in W(B)$ by the definition above. We now present the main Theorem concerning convergence of the GMRES algorithm.

Theorem 2.3:
If the linear system $B\boldsymbol{U} = \boldsymbol{c}$ is solved using GMRES then the mth residual, $\boldsymbol{r}_m = B\boldsymbol{U}_m - \boldsymbol{c}$, for $m \geq 0$ satisfies the following upper bound,

$$\frac{\|\boldsymbol{r}_m\|_2}{\|\boldsymbol{r}_0\|_2} \leq \sin^m \beta, \text{ where } \cos\beta = \frac{\text{dist}\,(0, W(B))}{\|B\|_2}. \tag{2.9}$$

The proof of this originally appeared in the PhD thesis of Elman [17]. The above Theorem tells one that if we can show that $\sin\beta < 1$ then the relative residual of the GMRES algorithm does indeed converge, i.e. $\frac{\|\mathbf{r}_m\|_2}{\|\mathbf{r}_0\|_2} \to 0$ as $m \to \infty$. The following Corollary to Theorem 2.3 shows one that this is indeed true under certain assumptions. Of course it is more interesting to make $\sin\beta$ as small as possible and then the convergence rate will be fast.

Corollary 2.4:
If we assume that $\|\mathbb{I} - B\|_2 \leq \sigma < 1$, where $\mathbb{I}$ is the $n \times n$ identity matrix then,

$$\cos\beta \geq \frac{1-\sigma}{1+\sigma}, \text{ and } \sin\beta \leq \frac{2\sqrt{\sigma}}{(1+\sigma)}.$$

Proof. Firstly if we assume that $\|I - B\|_2 \leq \sigma$ then,

$$\begin{aligned} \|B\|_2 &= \|-(\mathbb{I} - B) + \mathbb{I}\|_2, \text{ then using the triangle inequality,} \\ &\leq \|\mathbb{I} - B\|_2 + \|\mathbb{I}\|_2, \\ &\leq \sigma + 1. \end{aligned} \tag{2.10}$$

Then combining (2.9) and (2.10) gives the following,

$$\cos\beta = \inf_{\substack{\mathbf{x}\in\mathbb{C}^n \\ \|\mathbf{x}\|=1}} \frac{|\,(B\mathbf{x}, \mathbf{x})\,|}{1+\sigma},$$

We proceed by adding and subtracting $\mathbb{I}$ in the numerator which gives

$$\begin{aligned} \cos\beta &= \inf_{\substack{\mathbf{x}\in\mathbb{C}^n \\ \|\mathbf{x}\|=1}} \frac{|\,((B + \mathbb{I} - \mathbb{I})\,\mathbf{x}, \mathbf{x})\,|}{1+\sigma}, \\ &= \inf_{\substack{\mathbf{x}\in\mathbb{C}^n \\ \|\mathbf{x}\|=1}} \frac{|(\mathbf{x}, \mathbf{x}) - ((\mathbb{I} - B)\,\mathbf{x}, \mathbf{x})\,|}{1+\sigma} \end{aligned}$$

using the linearity of the inner product for the second line. Then if we use the reverse

triangle inequality and the Cauchy Schwarz inequality together,

$$\begin{aligned}\cos\beta &\geq \frac{1-\|\mathbb{I}-B\|_2}{1+\sigma},\\ &\geq \frac{1-\sigma}{1+\sigma}.\end{aligned}$$

Hence the result for $\cos\beta$ in (2.9). The result for $\sin\beta$ then follows immediately by squaring $\cos\beta$ and using the fact that $\sin^2\beta+\cos^2\beta=1$.

$$\begin{aligned}\sin^2\beta &\leq 1-\left(\frac{1-\sigma}{1+\sigma}\right)^2,\\ &= \frac{4\sigma}{(1+\sigma)^2}.\end{aligned}$$

Hence the result. □

The previous Corollary tells one that if $0\notin W(B)$, then $\cos\beta>0$ and hence $\sin\beta<1$, and therefore GMRES algorithm is guaranteed to converge. However it is not clear if this is true when solving the discretisation of the Helmholtz equation (2.2), i.e. when $B=A$.

In Table 2.1 we present the number of GMRES iterations to solve (2.2) to a tolerance of 10^{-6}. In Figure 2-1 we plot the Field of Values of the system matrix A from (2.2) for a fixed k using an algorithm for computing Field of Values found in [14]. In these experiments we increase k and fix the mesh spacing $h\sim k^{-\frac{3}{2}}$ and hence the total number of grid points $N\sim k^3$. For this example $\Omega=(0,1)^2$ and we set $f=1$, $g=0$ in (2.1). We also calculate the distance of the field of values, $W(A)$, from the origin to provide evidence to support the claim that $0\in W(A)$.

Table 2.1 and Figure 2-1 clearly show that $0\in W(A)$. But even though there is no theoretical justification that GMRES should converge it does. However as k is increased, and N accordingly, the number of GMRES iterations grows rapidly therefore an adequate preconditioner is required.

k	N	dist $(0,W(A))$	Iterations
5	324	0	35
10	2601	0	90
20	20449	0	264
40	162409	0	941

Table 2.1: *Number of GMRES iterations and CPU time for the solution of* (2.1) *with a fixed* k*, and total number of grid points* $N=n^2$*. Here* $n\sim k^{\frac{3}{2}}$*.*

Recently much research has focused on preconditioning with the so called *shifted*

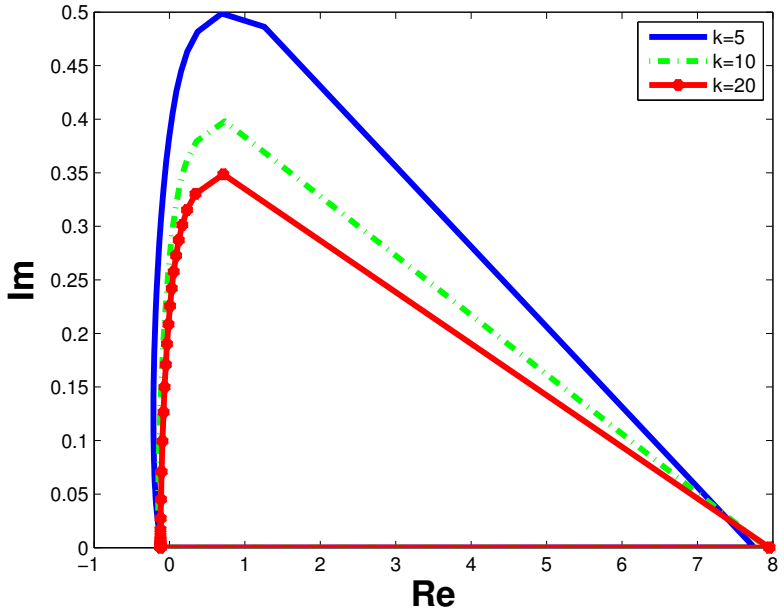

Figure 2-1: *The boundary of the field of values of A for $k = 5, 10, 20$.*

Laplace preconditioner. A brief review of previous literature is given in Section 2.2.1. This involves using the discretisation of the Helmholtz equation with a complex shift of $i\epsilon$ and impedance boundary condition,

$$\left.\begin{aligned} -\Delta u - k^2 u - i\epsilon u &= f, \text{ in } \Omega, \\ \frac{\partial}{\partial n} u - i\eta u &= g, \text{ on } \partial\Omega, \end{aligned}\right\} \tag{2.11}$$

where $\eta, \epsilon \in \mathbb{R}_+$ and all other variables are as previously defined for (2.1). It can be shown [38, Remark 2.2] that there exists a unique $u \in H^1(\Omega)$ which satisfies (2.11). Discretising (2.11) with low order finite elements, (which has been our convention) results in the linear system,

$$A_\epsilon \mathbf{U} = \mathbf{b}. \tag{2.12}$$

Before discussing preconditioning further we shall examine how the field of values and performance of GMRES changes when we introduce the absorbing term ϵ. We repeat the experiments of Table 2.1 in Tables 2.2, 2.3 and 2.4 where we increase $\epsilon = k, k^{\frac{3}{2}}, k^2$. All other parameters are as previously defined. The corresponding boundaries of the field of values are plotted in Figures 2-2, 2-3 and 2-4. What we observe from these results is that as ϵ is increased the bottom left corner of the boundary of the field of values is rotated away from zero and into the complex plane. This can be observed best from Figure 2-7 where $\epsilon = k^2$. The consequence for iterative solvers is that the

resultant linear system (2.12) is easier to solve. In Tables 2.2, 2.3 and 2.4 we list the number of GMRES iterations to solve (2.12). What can clearly be seen is that as ϵ increases the number of iterations decreases, with this fact being more apparent for larger k.

k	N	dist $(0, W(A_\epsilon))$	Iterations
5	324	0.048226	34
10	2601	0.015918	88
20	20449	0.006150	252
40	162409	0.001754	857

Table 2.2: *Number of GMRES iterations for the solution of* (2.1) *with a fixed k, and total number of grid points $N = n^2$. Here $n \sim k^{\frac{3}{2}}$ and $\epsilon = k$.*

k	N	dist $(0, W(A_\epsilon))$	Iterations
5	324	0.109159	33
10	2601	0.050287	82
20	20449	0.027501	220
40	162409	0.010651	557

Table 2.3: *Number of GMRES iterations for the solution of* (2.11) *with a fixed k, and total number of grid points $N = n^2$. Here $n \sim k^{\frac{3}{2}}$ and $\epsilon = k^{\frac{3}{2}}$.*

k	N	dist $(0, W(A_\epsilon))$	Iterations
5	324	0.241953	32
10	2601	0.159204	69
20	20449	0.122994	123
40	162409	0.094611	180

Table 2.4: *Number of GMRES iterations for the solution of* (2.11) *with a fixed k, and total number of grid points $N = n^2$. Here $n \sim k^{\frac{3}{2}}$ and $\epsilon = k^2$.*

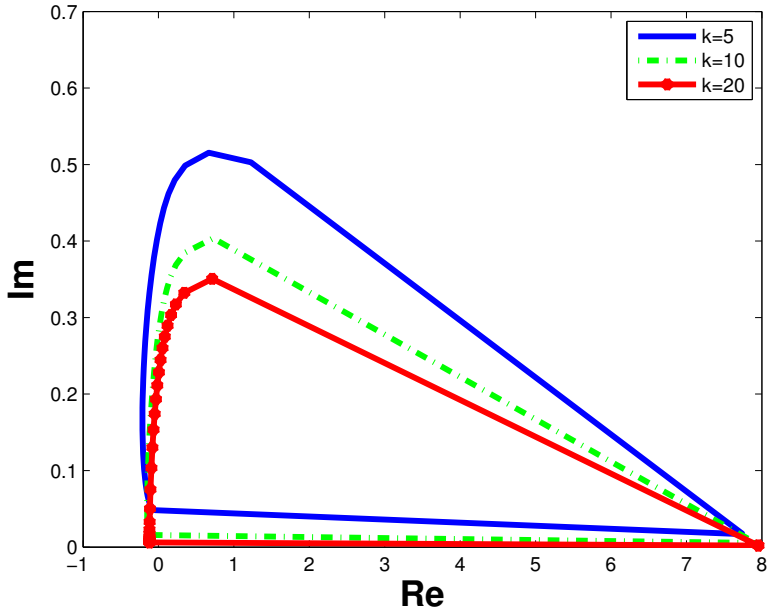

Figure 2-2: *The boundary of the field of values of* A_ϵ *for* $k = 5, 10, 20$ *with* $\epsilon = k$.

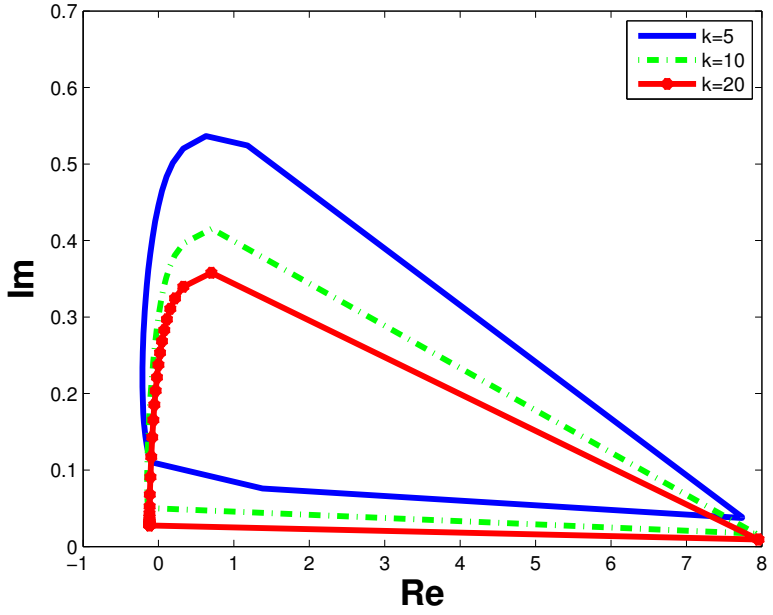

Figure 2-3: *The boundary of the field of values of* A_ϵ *for* $k = 5, 10, 20$ *with* $\epsilon = k^{\frac{3}{2}}$.

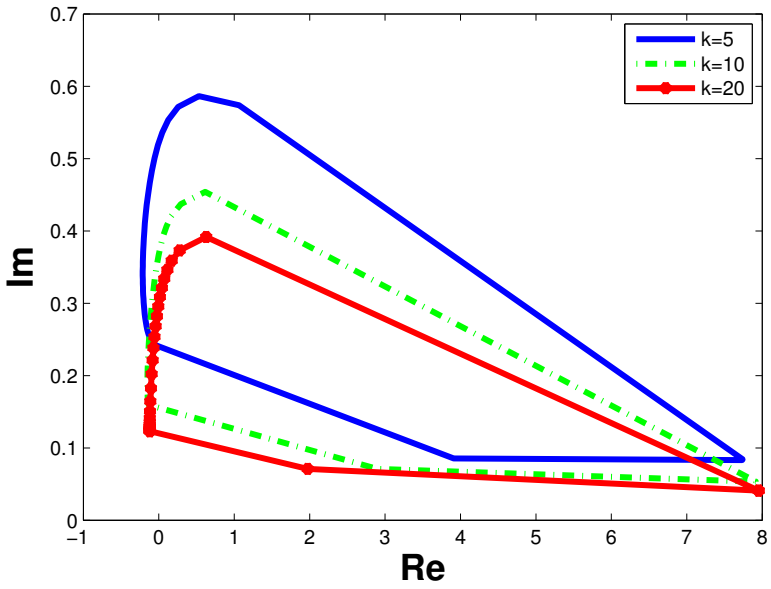

Figure 2-4: *The boundary of the field of values of* A_ϵ *for* $k = 5, 10, 20$ *with* $\epsilon = k^2$.

However, we are not interested in solving (2.12) but rather (2.2). Therefore we precondition (2.2) by pre-multiplying (known as left preconditioning) the linear system (2.2) by a suitable matrix P, or more commonly the action of P generated by some numerical method. That is we solve

$$PA\mathbf{U} = P\mathbf{b}. \tag{2.13}$$

We could equally solve instead

$$AP\mathbf{V} = \mathbf{b}$$

where $\mathbf{U} = P\mathbf{V}$ which is known as right preconditioning. The optimal choice would of course be to choose $P = A^{-1}$ as $A^{-1}A = \mathbb{I}$. But of course it is not practical to use A^{-1} itself as a preconditioner in (2.13), as the inversion of this matrix would be as expensive as solving the original system (2.2). We instead use some approximation of A^{-1}, which we shall denote M^{-1}. The hope is that M^{-1} is both cheap to compute and *close* to A^{-1}, i.e. $M^{-1}A \approx \mathbb{I}$. Many authors have chosen to use an approximation of A_ϵ^{-1}. The reason is that A_ϵ^{-1} is close to A^{-1} for small enough ϵ, and solving the linear system (2.12) is easier than solving (2.2). However the latter becomes increasingly true for ϵ large enough so there may be conflicting demands on the choice of ϵ. Therefore we construct a preconditioner M_ϵ^{-1} which involves approximating A_ϵ^{-1} by some numerical method. The preconditioned system that we solve is,

$$M_\epsilon^{-1}A\mathbf{U} = M_\epsilon^{-1}\mathbf{b}. \tag{2.14}$$

Recalling 2.4 and replacing B with $M_\epsilon^{-1}A$ we can see that for GMRES to perform well we require that $\|\mathbb{I} - M_\epsilon^{-1}A\| \leq \sigma < 1$. But how do we choose ϵ for GMRES to converge independently of k?

A recent paper by Gander, Graham and Spence [38] addresses this issue. If we write,

$$\begin{aligned}\|\mathbb{I} - M_\epsilon^{-1}A\|_2 &= \|\mathbb{I} - M_\epsilon^{-1}A_\epsilon + M_\epsilon^{-1}A_\epsilon\left(\mathbb{I} - A_\epsilon^{-1}A\right)\|_2, \\ &\leq \|\mathbb{I} - M_\epsilon^{-1}A_\epsilon\|_2 + \|M_\epsilon^{-1}A_\epsilon\|_2\|\mathbb{I} - A_\epsilon^{-1}A\|_2. \end{aligned} \tag{2.15}$$

Therefore we can see from (2.15) that a sufficient condition for M_ϵ^{-1} to be a good preconditioner for A is that both $\|\mathbb{I} - M_\epsilon^{-1}A_\epsilon\|$ and $\|\mathbb{I} - A_\epsilon^{-1}A\|$ are small. However this requires that we satisfy the two following conditions,

(1) That A_ϵ^{-1} be an effective preconditioner for A, so $\|\mathbb{I} - A_\epsilon^{-1}A\|_2$ is small,

(2) and that M_ϵ^{-1} be an effective preconditioner for A_ϵ, so $\|\mathbb{I} - M_\epsilon^{-1}A_\epsilon\|_2$ is small.

This unfortunately poses two conflicting requirements on the value of ϵ. If (1) is to be satisfied then ϵ should be small, as the ideal preconditioner for A is A_0^{-1} i.e. $\epsilon = 0$. However for (2) to be satisfied one requirement is that M_ϵ^{-1} is cheap to construct. We can see from from figures 2-2, 2-3 and 2-4 that as ϵ increases the field of values gets further away from the origin. Then Theorem 2.3 tells us that as $dist(0, W(A)) > 0$ then GMRES will converge faster when solving (2.12) when ϵ increases.

In [38] a rigorous analysis of the first requirement is provided. The main results of this paper are the following.

Theorem 2.5:
If Ω is a convex polygon or star shaped with respect to a ball (see definition in [38]) and that A and A_ϵ are formed using low order finite elements on a quasi-uniform mesh (again defined in [38]). Assume that $\frac{\epsilon}{k^2}$ is bounded and $\eta = k$. Then given $k_0 > 0$ and $C>0$, there exist $C_1, C_2>0$ (and independent of k, ϵ and the finite element mesh spacing h) such that if $hk^2 \geq C$ and $hk\sqrt{|k^2 - \epsilon|} \leq C_2$, then

$$\|\mathbb{I} - A_\epsilon^{-1} A\|_2 \leq C_1 \frac{\epsilon}{k} \tag{2.16}$$

for all $k \geq k_0$.

Remark 2.6:
Theorem 2.5 holds if $\eta = k$ is replaced with $\eta = \sqrt{k^2 - i\epsilon}$ in (2.11).

Theorem 2.7:
If the assumptions of Theorem 2.5 hold, and $\frac{\epsilon}{k}$ is sufficiently small, then GMRES solves $A_\epsilon^{-1} A \boldsymbol{U} = A_\epsilon^{-1} \boldsymbol{b}$ in a number of iterations which is independent of k.

Therefore this tells one that if $\frac{\epsilon}{k}$ is chosen sufficiently small then A_ϵ^{-1} acts as an optimal preconditioner for A. The proofs of Theorems 2.5 and 2.7 can be found in §4 of [38].

In Table 2.5, 2.6, 2.7 we give GMRES iteration counts and distance of the field of values from the origin for the preconditioned system $A_\epsilon^{-1} A\mathbf{U} = A_\epsilon^{-1}\mathbf{1}$. Boundaries of the field of values in each case are given in Figures 2-5, 2-6, 2-7. We observe that for increasing k the boundary of the field of values stays bounded away from the origin and close to 1. This is verified by Figure 2-5. As a result of this we can see that the number of GMRES iterations stays constant as k increases as predicted by Theorem 2.5. However if ϵ is increased to $\epsilon \sim k^{\frac{3}{2}}$ or $\epsilon \sim k^2$ say, then the k independent convergence of GMRES is lost and in fact $\frac{\epsilon}{k}$ now grows with k. Indeed the number of GMRES iterations now grows with k for this increased choice of ϵ. This can be seen in the results of Tables 2.6 and 2.7. When $\epsilon = k^{\frac{3}{2}}$ the number of iterations increases roughly as $\log k$ and if $\epsilon = k^2$ the iterations increase linearly with k. This behaviour is expected as the boundary of the field of values gets closer to 0 as ϵ increases as is shown in Tables 2.6 and 2.7 and

in Figures 2-6 and 2-7.

The problem remains to compute a good cheap approximation for A_ϵ^{-1} for given ϵ and that is one main focus of this thesis. In the next section of this chapter we outline a multiplicative domain decomposition method for the iterative solution of (2.12). The analysis of this algorithm is given in the following Chapter 3 and numerical experiments of its use as an iterative method and as a preconditioner for A_ϵ and A in Chapter 4.

k	$N = n^2$	dist $\left(0, W(A_\epsilon^{-1}A)\right)$	Iterations
5	324	0.761364	5
10	2601	0.754029	6
20	20449	0.745377	6
40	162409	0.736547	6

Table 2.5: *Number of GMRES iterations and CPU time for the solution of* $A_\epsilon^{-1}A = A_\epsilon^{-1}\mathbf{1}$ *with a fixed* k*, and total number of grid points* N*. Here* $\epsilon = k$ *and* $n \sim k^{\frac{3}{2}}$

k	$N = n^2$	dist $\left(0, W(A_\epsilon^{-1}A)\right)$	Iterations
5	324	0.541273	6
10	2601	0.453691	8
20	20449	0.338737	11
40	162409	0.227651	14

Table 2.6: *Number of GMRES iterations and CPU time for the solution of* $A_\epsilon^{-1}A = A_\epsilon^{-1}\mathbf{1}$ *with a fixed* k*, and total number of grid points* N*. Here* $\epsilon = k^{\frac{3}{2}}$ *and* $n \sim k^{\frac{3}{2}}$

k	$N = n^2$	dist $\left(0, W(A_\epsilon^{-1}A)\right)$	Iterations
5	324	0.287407	7
10	2601	0.142148	13
20	20449	0.049498	24
40	162409	0.014265	48

Table 2.7: *Number of GMRES iterations and CPU time for the solution of* $A_\epsilon^{-1}A = A_\epsilon^{-1}\mathbf{1}$ *with a fixed* k*, and total number of grid points* N*. Here* $\epsilon = k^2$ *and* $n \sim k^{\frac{3}{2}}$

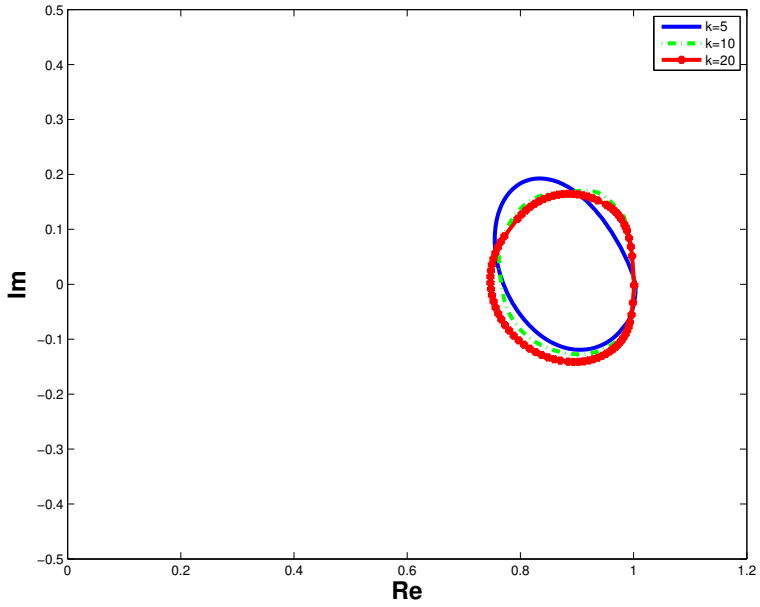

Figure 2-5: *The boundary of the field of values of* $A_\epsilon^{-1}A$ *for* $k = 5, 10, 20$*, with* $\epsilon = k$*.*

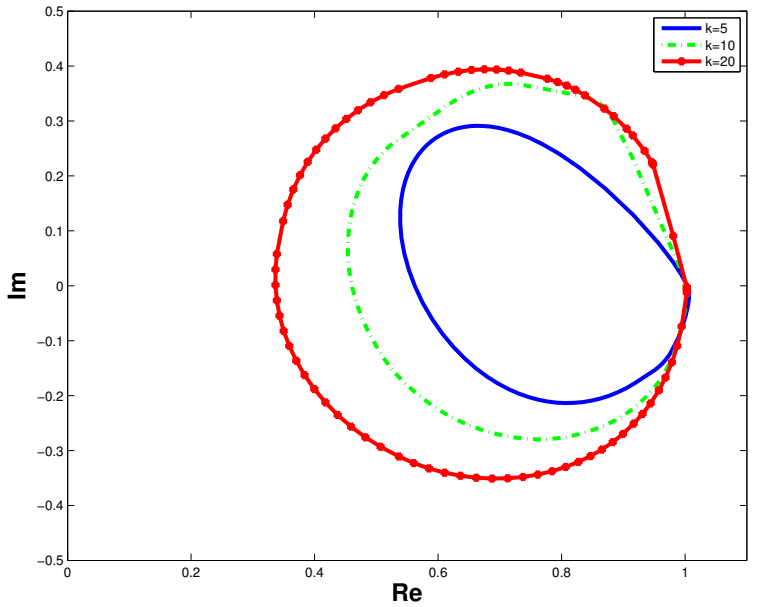

Figure 2-6: *The boundary of the field of values of* $A_\epsilon^{-1}A$ *for* $k = 5, 10, 20$*, with* $\epsilon = k^{\frac{3}{2}}$*.*

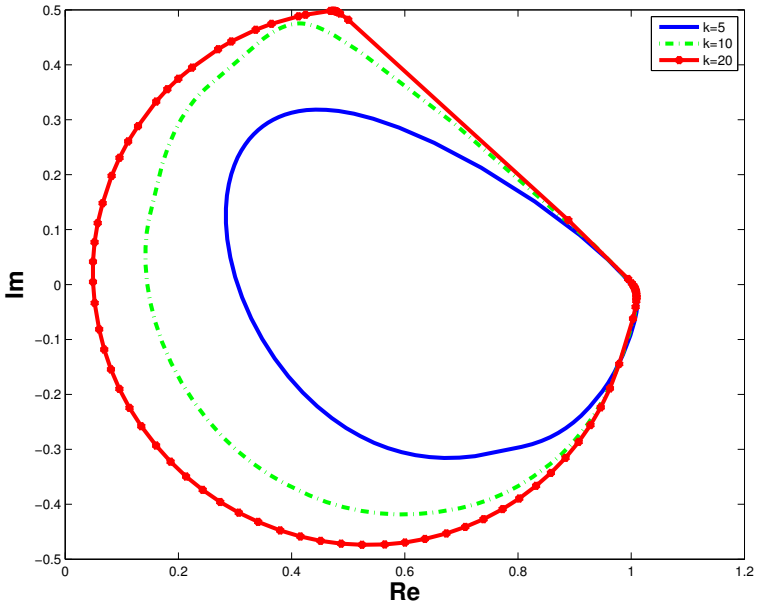

Figure 2-7: *The boundary of the field of values of* $A_\epsilon^{-1}A$ *for* $k = 5, 10, 20$*, with* $\epsilon = k^2$.

2.2.1 Previous work on *shifted Laplace* preconditioners

We now detail some previous literature concerned with the use of so called *shifted Laplace preconditioners*. We give details of their choice of ϵ and the different methods used for forming M_ϵ^{-1}.

The earliest known work was proposed in [1]. In this paper the authors used a preconditioner which was formed by an approximation of $-\Delta^{-1}$ using a single sweep of SSOR [51], a variant of Gaussian elimination. In [58] the authors construct a preconditoner by approximating $\left(-\Delta - \alpha k^2\right)^{-1}$ with $\alpha \in \mathbb{C}$ and $Im\left(\alpha\right) < 0$ from one multigrid V-cycle, or an incomplete ILU factorisation. The resulting spectral properties of $M_\alpha^{-1}A$ were studied numerically. This choice of α, which is equivalent to $\epsilon \sim k^2$ in our notation, was also used in another paper by the same authors [59]. The choice of $\epsilon \sim k^2$ used here was motivated by an eigenvalue analysis of a 1D model Helmholtz problem with Dirichlet boundary conditions.

Another study of multigrid for Helmholtz can be found in [44]. Here the authors use multigrid to solve problems involving A_ϵ. These authors include a Fourier analysis of multigrid with the conclusion that $\epsilon \sim k^2$ is needed for multigrid to be a good solver for A_ϵ. Also included is an eigenvalue analysis of a finite difference discretisation of A_ϵ. This showed that if one chooses $\epsilon < k$ then the resultant clustering of the eigenvalues of $A_\epsilon^{-1}A$ is close to 1. Therefore this is favourable for a Krylov method, and is the equivalent to the observations made in [38].

Shifted Laplace precondtioners have also been used for other methods. In [31] one iteration of a restricted additive Schwarz (RAS) domain decomposition method was used to approximate A_ϵ^{-1} with a choice of $\epsilon \sim k^2$ in M_ϵ^{-1}. The motivation here was mainly through numerical experimentation. Finally the sweeping preconditioner of Engquist and Ying [5], [6] replaces k^2 with $(k + i\alpha)^2$ with $\alpha \in \mathbb{R}$ in the formation of M_α^{-1}. If we expand $(k + i\alpha)^2 = k^2 + 2i\alpha k - \alpha^2$. Then as α is a constant independent of k this can be seen to correspond to a choice of $\epsilon \sim k$ in our own notation. This sweeping preconditioner has proved very effective with numerical evidence of k independent convergence for some numerical examples. We discuss this method in detail in Chapter 5.

2.3 The multiplicative Schwarz algorithm with overlap

In the following we develop an optimised Robin type transmission condition for the Schwarz domain decomposition method when used as an iterative solver for the Helmholtz equation with an absorbing term ϵ. The analysis is on a simplified model case with two subdomains. This method can also be used as a preconditioner by using a finite number of iterates to approximate A_ϵ^{-1}. Firstly the algorithm is presented and then a Fourier analysis performed to derive the corresponding convergence rate of the Schwarz algorithm. We then detail some possible low order choices of transmission condition and show how these influence the convergence of the algorithm.

2.3.1 Review of previous work on optimised Schwarz methods for Helmholtz problems

There has been much literature dedicated to optimised Schwarz methods for solving the Laplace equation and the Helmholtz equation. The main motivation for looking at optimised methods is that the alternating Schwarz method [8] (where Dirichlet boundary conditions are used on the interfaces) does not converge in general for large k when applied to the Helmholtz equation (2.1). Nonetheless, in [10] the Additive Schwarz algorithm (a variant of the classical alternating method, see [8]) was applied to the Helmholtz equation with a fine coarse space so as to ensure convergence. Unfortunately as the coarse space was required to be almost as fine as the fine grid it was not practical for large k. Another early application of the classical alternating algorithm to Helmholtz problems (in fact, more generally, to Maxwell's equations) was in [3] where the Dirichlet interface conditions of the classical alternating method were changed to those of Robin type. It was found that this different choice of interface condition leads to a domain decomposition method which was convergent. This method was then employed also in the following papers [4], [15], [30] among others. The idea of using Robin type interface conditions actually pre-dates the previous authors and was introduced for the non-overlapping Schwarz method by Lions in [35]. Here the author mentions that the constants in the Robin interface conditions could be replaced by functions or local or non-local operators. This statement, at the end of the paper, seems to be the first mention of what later became known as *optimised Schwarz methods*. What has followed in the last decade has seen the optimised Schwarz methods applied to many PDE problems including the Helmholtz and Maxwell problems. We give the following references specific to the Helmholtz problem; for the non-overlapping method [21], [39], [40] and for the overlapping method [22]. The main advantages of optimised Schwarz methods are:

- That they do not require overlap to converge, and are guaranteed to converge quicker than the classical Schwarz method i.e. Dirichlet interface conditions.

- They are relatively simple to implement into existing Schwarz solvers. One needs to only change the part of the code which deals with the exchange of data on domain interfaces.

- The formulation of the method requires the solution of relatively simple optimisation problems which can be solved quickly numerically. Though in some cases closed form solutions can exist in certain asymptotic regimes, see Chapter 3.

In this following chapter we shall outline the optimised Schwarz method for the solution of the Helmholtz equation (2.1) with an absorbing term ϵ . A Fourier analysis shall then be performed on the algorithm (2.17) below to show that it is indeed convergent.

2.3.2 The multiplicative Schwarz method for the solution of the Helmholtz equation with absorption

We now consider using the two subdomain Multiplicative Schwarz algorithm to solve (2.11) iteratively. Our domain Ω is assumed to be a rectangle and is decomposed into two overlapping rectangular subdomains Ω_1 and Ω_2 as shown in Figure 2-8, where L denotes the overlap parameter. The multiplicative Schwarz algorithm is then the

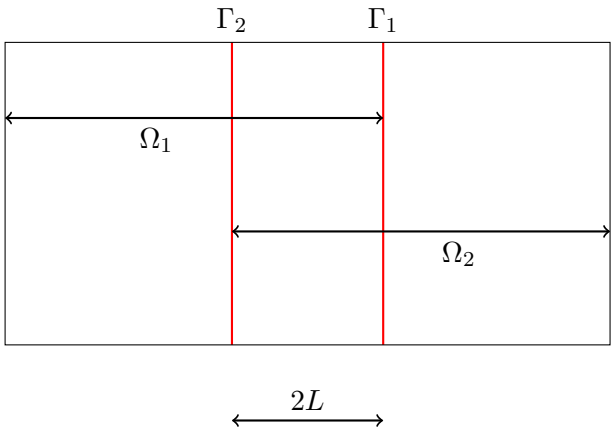

Figure 2-8: *Cartoon of the decomposition* $\Omega = (0,1)^2$ *into two overlapping subdomains* Ω_1 *and* Ω_2.

following at iterate n, where the iteration is started with an initial guess u_1^0,

$$\left(-\Delta - k^2 + i\epsilon\right) u_1^n(x,y) = f(x,y), \text{ in } \Omega_1 \tag{2.17a}$$

$$(\partial_x + S)u_1^n(x,y) = (\partial_x + S)u_2^{n-1}(x,y), \text{ on } \Gamma_1 \tag{2.17b}$$

$$\left(-\Delta - k^2 + i\epsilon\right) u_2^{n+1}(x,y) = f(x,y), \text{ in } \Omega_2 \tag{2.17c}$$

$$(-\partial_x + S)u_2^{n+1}(x,y) = (-\partial_x + S)u_1^n(x,y), \text{ on } \Gamma_2. \tag{2.17d}$$

Remark 2.8:
The algorithm (2.17) *actually solves iteratively the adjoint of the PDE problem which we had mentioned previously* (2.11)*, namely*

$$\begin{aligned} -\Delta u - k^2 u + i\epsilon u &= f, \text{ in } \Omega, \\ \frac{\partial}{\partial n} u + i\eta u &= g, \text{ on } \partial\Omega. \end{aligned}$$

We make the reader aware that we consider solving the adjoint problem as it results in a simpler analysis in Chapter 3, and that all of the main results presented in this thesis for Schwarz methods are the same for the problem (2.11)*.*

It is easy to see that u will satisfy (2.17b) and (2.17d).If we remind ourselves that u satisfies (2.11) then $u \in H^1(\Omega)$. Hence u and its first derivative are continuous and therefore $u_1 = u_2$ on Γ_i and $\partial_x u_1 = \partial_x u_2$ on Γ_i, for $i = 1, 2$. Therefore u must satisfy the interface conditions (2.17b) and (2.17d). In the following we assume that the operator S is a linear operator acting tangentially along the interfaces Γ_1, Γ_2, and that it is diagonalisable by a Fourier transformation in the y direction (see (2.18)). To start we define the Fourier transform.

Definition 2.9:
The Fourier transform, $\hat{f}(\xi)$ of a function $f(x) : \mathbb{R} \to \mathbb{R}$, where $f \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$, is defined by,

$$\begin{aligned} \hat{f}(\xi) &= (\mathcal{F}f)[\xi] \\ &=: \int_{-\infty}^{\infty} e^{-i\xi y} f(y) dy \end{aligned}$$

and the inverse Fourier transform defined by

$$\begin{aligned} f(y) &= \left(\mathcal{F}^{-1}\hat{f}\right)[y] \\ &=: \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\xi y} \hat{f}(\xi) d\xi \end{aligned}$$

Referring back to (2.17) we assume that S has the property that for all $\phi \in L^1(\mathbb{R}) \cap$

$L^2(\mathbb{R})$,

$$\left(\widehat{S\phi}\right)(\xi) = \sigma(\xi)\hat{\phi}(\xi). \tag{2.18}$$

for some complex scalar function $\sigma := \sigma(\xi)$ usually called the *symbol* of S. Let us then consider the error in the Schwarz algorithm (2.17) at iterate n. This is defined as,

$$E_j^n(x,y) = \left(u_j - u_j^n\right)(x,y), \ (x,y) \in \Omega_j \ \text{ for } \ j = 1,2, \tag{2.19}$$

where u_j is the solution of (2.11) on Ω_j and u_j^n is the n^{th} approximate solution on Ω_j of the Schwarz algorithm (2.17). This allows one to write an iteration for the error as,

$$\left(-\Delta - k^2 + i\epsilon\right) E_1^n(x,y) = 0, \text{ in } \Omega_1 \tag{2.20a}$$

$$(\partial_x + S)E_1^n(x,y) = (\partial_x + S)E_2^{n-1}(x,y), \text{on } \Gamma_1 \tag{2.20b}$$

$$\left(-\Delta - k^2 + i\epsilon\right) E_2^{n+1}(x,y) = 0, \text{ in } \Omega_2 \tag{2.20c}$$

$$(-\partial_x + S)E_2^{n+1}(x,y) = (-\partial_x + S)E_1^n(x,y), \text{on } \Gamma_2, \tag{2.20d}$$

We want to know how fast the error decays to zero in (2.20). We now derive an (approximate) expression for the Fourier transform of the error. The first step is to (for purposes of the analysis) replace Ω with all of $\mathbb{R}^2$ this allows one to take the Fourier transform of (2.20) and perform a Fourier analysis. It is possible to perform a discrete Fourier analysis rather than a continuous analysis, we choose the continuous analysis as it is simpler. The following result is new but uses the similar ideas as that of [39].

Theorem 2.10:
After Fourier transform, the Schwarz algorithm (2.20) *satisfies,*

$$\hat{E}_j^n(x,\xi) = \rho(\xi,k,\epsilon,\sigma,L)\hat{E}_j^{n-2}(x,\xi), \ \textit{for } j = 1,2. \tag{2.21}$$

where the convergence rate ρ is given by

$$\rho(\xi,k,\epsilon,\sigma,L) = \left(\frac{-\lambda(\xi,k,\epsilon)+\sigma}{\lambda(\xi,k,\epsilon)+\sigma}\right)^2 e^{-2\lambda(\xi,k,\epsilon)L}, \tag{2.22}$$

with,

$$\lambda(\xi,k,\epsilon) = \sqrt{\xi^2 - k^2 + i\epsilon}.$$

Proof. A Fourier transform in the y direction is performed on the algorithm (2.20).

Integrating (2.20a) and (2.20c) over $y \in (-\infty, \infty)$ we have

$$\int_{-\infty}^{\infty} e^{-i\xi y}(-\partial_{xx}^2 - \partial_{yy}^2 - k^2 + i\epsilon)E_i^n(x,y)dy = 0. \tag{2.23}$$

Integrating by parts then one can easily show that

$$\left(\frac{\partial^2 E_i^n(x,y)}{\partial y^2}\right)^{\wedge}(\xi) = -\xi^2 \widehat{E}_i^n(x,\xi), \text{ for } i = 1,2. \tag{2.24}$$

Then inserting (2.24) into (2.23) it follows that,

$$\int_{-\infty}^{\infty} e^{-i\xi y}(-\partial_{xx}^2 + \xi^2 - k^2 + i\epsilon)E_i^n(x,y)dy = 0, \text{ for } i = 1,2.$$

Which we can write more concisely as the following ODE in x,

$$(-\partial_{xx}^2 + \xi^2 - k^2 + i\epsilon)\widehat{E}_i^n(x,\xi) = 0, \text{ for } i = 1,2. \tag{2.25}$$

Inserting into the Fourier transform of (2.20) we obtain

$$(-\partial_{xx}^2 + \xi^2 - k^2 + i\epsilon)\widehat{E}_1^n(x,\xi) = 0, \ x < L, \ \xi \in \mathbb{R}, \tag{2.26a}$$

$$(\partial_x + \sigma)\widehat{E}_1^n(x,\xi) = (\partial_x + \sigma)\widehat{E}_2^{n-1}(x,\xi), \ x = L, \tag{2.26b}$$

$$(-\partial_{xx}^2 + \xi^2 - k^2 + i\epsilon)\widehat{E}_2^{n+1}(x,\xi) = 0, \ x > 0, \ \xi \in \mathbb{R}, \tag{2.26c}$$

$$(-\partial_x + \sigma)\widehat{E}_2^{n+1}(x,\xi) = (-\partial_x + \sigma)\widehat{E}_1^n(x,\xi), \ x = 0, \tag{2.26d}$$

where we recall that σ is a scalar multiplier arising from the Fourier transform of the operator S as previously in (2.18). It is clear to see that the ODEs (2.26a) and (2.26c) have general solutions of the form,

$$\widehat{E}_j^n(x,\xi) = A_j e^{\lambda(\xi,k,\epsilon)x} + B_j e^{-\lambda(\xi,k,\epsilon)x}. \ j = 1,2, \tag{2.27}$$

for some $\lambda(\xi,k,\epsilon) \in \mathbb{C}$ satisfying the characteristic equation:

$$\lambda^2 = \xi^2 - k^2 + i\epsilon.$$

We then choose

$$\lambda(\xi,k,\epsilon) = \sqrt{\xi^2 - k^2 + i\epsilon}. \tag{2.28}$$

In order to describe which square root is under consideration we write λ as

$$\lambda(\xi,k,\epsilon) = \lambda_R(\xi,k,\epsilon) + i\lambda_I(\xi,k,\epsilon), \tag{2.29}$$

where $\lambda_R(\xi,k,\epsilon)$, $\lambda_I(\xi,k,\epsilon)$ are the real and imaginary parts of $\lambda(\xi,k,\epsilon)$ respectively. Then as we have chosen $\epsilon > 0$ it follows immediately that $\mathrm{Im}\left(\xi^2 - k^2 + i\epsilon\right) > 0$, and therefore that $\xi^2 - k^2 + i\epsilon$ must lie in either the first or second quadrant and not on the real line. We adopt the convention that the square root in (2.29) is taken to be the one lying in the first quadrant. Hence it must be true that,

$$\lambda_R(\xi,k,\epsilon) > 0, \text{ and } \lambda_I(\xi,k,\epsilon) > 0. \tag{2.30}$$

We recall that as we have taken the Fourier transform to get (2.26) it is necessary that

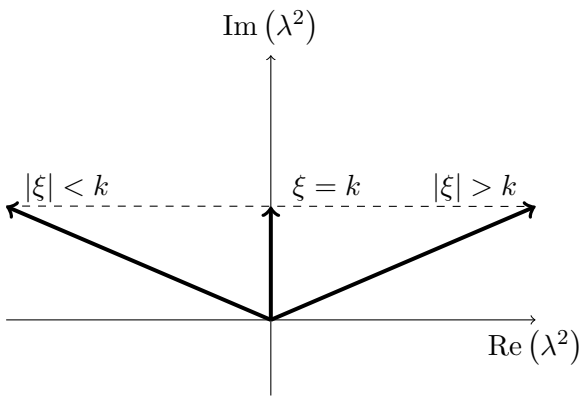

Figure 2-9: *Cartoon of $\lambda^2(\xi,k,\epsilon)$ (**bold line**) in the plane.*

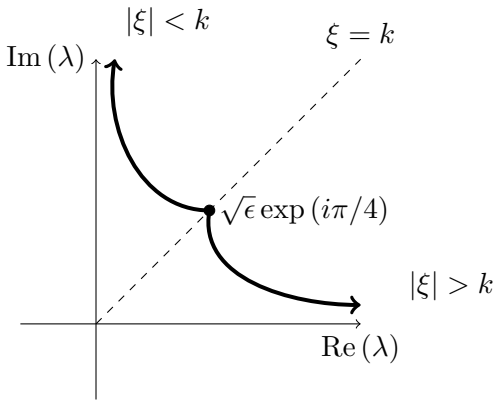

Figure 2-10: *Cartoon of $\lambda(\xi,k,\epsilon)$ (**bold line**) in the plane.*

(2.27) must decay as $x \to \pm\infty$. For example on Ω_1 we have the general solution,

$$\hat{E}_1^n(x,\xi) = A_1 e^{\lambda(\xi,k,\epsilon)x} + B_1 e^{-\lambda(\xi,k,\epsilon)x}. \tag{2.31}$$

This general solution for the error on Ω_1 should decay to zero as $x \to -\infty$. If we recall that $\lambda = \lambda_R + i\lambda_I$ where $\lambda_R, \lambda_I > 0$ then the above general solution will have oscillatory terms $e^{\pm i\lambda_I x}$ and terms $e^{\pm\lambda_R x}$ which will either increase or decrease exponentially. Therefore as $e^{\lambda_R x} \to 0$ and in addition $e^{\lambda_R x} \to \infty$ as $x \to -\infty$ this indicates that we should choose $B_1 = 0$ in (2.31). From this one obtains that the general solution is given by $\hat{E}_1^n(x,\xi) = A_1 e^{\lambda(\xi,k,\epsilon)x}$. Then setting $x = L$ gives $\hat{E}_1^n(L,\xi) = A_1 e^{\lambda(\xi,k,\epsilon)}$ and hence,

$$\hat{E}_1^n(x,\xi) = \hat{E}_1^n(L,\xi) e^{\lambda(\xi,k,\epsilon)(x-L)}, \ x \le L, \tag{2.32}$$

and analogously in Ω_2 the solution is given by

$$\hat{E}_2^{n+1}(x,\xi) = \hat{E}_2^{n+1}(0,\xi) e^{-\lambda(\xi,k,\epsilon)x}, \ x \ge 0, \tag{2.33}$$

We now proceed by inserting (2.32) and (2.33) into (2.26) and look at what we obtain for a single step of the Schwarz algorithm. Starting for example with the boundary

condition when $x = L$ we get,

$$(\lambda(\xi,k,\epsilon)+\sigma_1)\hat{E}_1^n(L,\xi) = (-\lambda(\xi,k,\epsilon)+\sigma_1)\hat{E}_2^{n-1}(L,\xi). \tag{2.34}$$

Rearranging this one obtains,

$$\hat{E}_1^n(L,\xi) = \left(\frac{-\lambda(\xi,k,\epsilon)+\sigma}{\lambda(\xi,k,\epsilon)+\sigma}\right)\hat{E}_2^{n-1}(L,\xi).$$

Then using (2.33) with $x = L$ gives,

$$\hat{E}_1^n(L,\xi) = \left(\frac{-\lambda(\xi,k,\epsilon)+\sigma}{\lambda(\xi,k,\epsilon)+\sigma}\right)\hat{E}_2^{n-1}(0,\xi)e^{-\lambda(\xi,k,\epsilon)L}. \tag{2.35}$$

Similarly one can obtain the following for the boundary condition at $x = 0$,

$$\hat{E}_2^{n+1}(0,\xi) = \left(\frac{-\lambda(\xi,k,\epsilon)+\sigma}{\lambda(\xi,k,\epsilon)+\sigma}\right)\hat{E}_1^n(L,\xi). \tag{2.36}$$

Combining (2.35) and (2.36) we obtain,

$$\hat{E}_1^{n+1}(L,\xi) = \left(\frac{-\lambda(\xi,k,\epsilon)+\sigma}{\lambda(\xi,k,\epsilon)+\sigma}\right)^2\hat{E}_1^{n-1}(L,\xi)e^{-2\lambda(\xi,k,\epsilon)L},$$

We then define the convergence rate as

$$\rho(\xi,k,\epsilon,\sigma,L) = \left(\frac{-\lambda(\xi,k,\epsilon)+\sigma}{\lambda(\xi,k,\epsilon)+\sigma}\right)^2 e^{-2\lambda(\xi,k,\epsilon)L}.$$

Therefore the n^{th} iterate on subdomain is written as,

$$\hat{E}_j^n(x,\xi) = \rho(\xi,k,\epsilon,\sigma,L)\hat{E}_j^{n-2}(x,\xi),\ j = 1,2,$$

with ρ as given in (2.22). □

The following Corollary shows that choosing $\sigma(\xi,k,\epsilon) = \lambda(\xi,k,\epsilon)$ is the optimal choice in the sense that the Schwarz algorithm (2.17) converges after one iteration on each subdomain.

Corollary 2.11:
If $\sigma(\xi) = \lambda(\xi,k,\epsilon)$ and S is defined as (2.18) *then the Schwarz algorithm* (2.17) *converges in two iterations for all $\xi \in [0,\infty)$.*

Proof. One simply inserts the choice of σ into (2.22). For example

$$\rho(\xi, k, \epsilon, \sigma, L) = \left(\frac{-\lambda(\xi, k, \epsilon) + \lambda(\xi, k, \epsilon)}{\lambda(\xi, k, \epsilon) + \lambda(\xi, k, \epsilon)} \right)^2 e^{-2\lambda(\xi,k,\epsilon)L} = 0.$$

Then recalling (2.21)

$$\begin{aligned} \hat{E}_j^2(0, \xi) &= \rho(\xi, k, \epsilon, \sigma, L)\hat{E}_j^0(0, \xi), \ j = 1, 2, \\ &= 0, \ \text{as } \rho(\xi, k, \epsilon) = 0. \end{aligned}$$

Therefore the Schwarz algorithm (2.17) will converge in two iterations, independent of the initial guess. □

However the choice of σ in Corollary 2.11 is not very practical in terms of implementation as it is very expensive to apply the action of the resulting operator S as required in (2.17). If we recall (2.18) and take the inverse Fourier transform of both sides this gives,

$$S\phi(\xi) = \mathcal{F}^{-1}\left(\sigma(\xi)\hat{\phi}(\xi)\right).$$

We can then see that to apply S to a function $\phi(\xi)$ this involves firstly computing the Fourier transform of ϕ, and then taking the inverse Fourier transform of the product of σ and $\hat{\phi}$. If $\sigma(\xi) := \lambda(\xi, k, \epsilon)$ then this is computationally very expensive, as each time we do we have to compute integrals for the Fourier transform and its inverse. This is true even with the FFT [13], but these would have to be applied for every grid point on the interfaces Γ_1 and Γ_2 leading to full matrices which is not suitable for a direct solver.

We now seek out approximations to S which are cheaper. What we find is that one can obtain a convergent algorithm without overlap by simply Taylor expanding $\lambda(\xi, k, \epsilon)$ as a function of ξ and using the lowest order term. This is the topic of §2.4.1. However, one can do even better by solving an optimisation problem involving the convergence rate (2.22). This is explained in detail in Chapter 3.

We now investigate the behaviour of the real and imaginary parts of $\lambda(\xi, k, \epsilon)$. Firstly we show that for fixed ξ, k and ϵ both $\lambda_R(\xi, k, \epsilon)$ and $\lambda_I(\xi, k, \epsilon)$ are strictly positive. We then derive some ODEs involving $\lambda_R(\xi, k, \epsilon)$, $\lambda_I(\xi, k, \epsilon)$ and their derivatives (with respect to ξ) which will prove to be very useful in the analysis in Chapter 3. Finally we calculate asymptotic formulas for $\lambda_R(\xi, k, \epsilon)$ and $\lambda_I(\xi, k, \epsilon)$ at fixed values of ξ for k increasing.

2.3.3 Some elementary results about $\lambda_R(\xi, k, \epsilon)$, $\lambda_I(\xi, k, \epsilon)$

We start by proving some results about $\lambda_R(\xi, k, \epsilon)$, $\lambda_I(\xi, k, \epsilon)$, as these functions will be used throughout the following analysis in Chapter 3. Let us recall the definition of $\lambda(\xi, k, \epsilon)$,

$$\lambda(\xi, k, \epsilon) = \sqrt{\xi^2 - k^2 + i\epsilon},$$

which we rewrite in its real and imaginary parts as,

$$\lambda(\xi, k, \epsilon) = \lambda_R(\xi, k, \epsilon) + i\lambda_I(\xi, k, \epsilon).$$

In Figure 2-11 we have plotted the real and imaginary parts of $\lambda(\xi, k, \epsilon)$ for $k = \epsilon = \pi$ and $\xi \in [1, 2k]$. Once again we remind the reader that we have taken the convention of choosing $\lambda(\xi, k, \epsilon)$ in the first quadrant. The first thing to notice is that when $\xi = k$ then the real and imaginary parts of $\lambda(\xi, k, \epsilon)$ are equal. This is represented in the plot by the red asterisk.

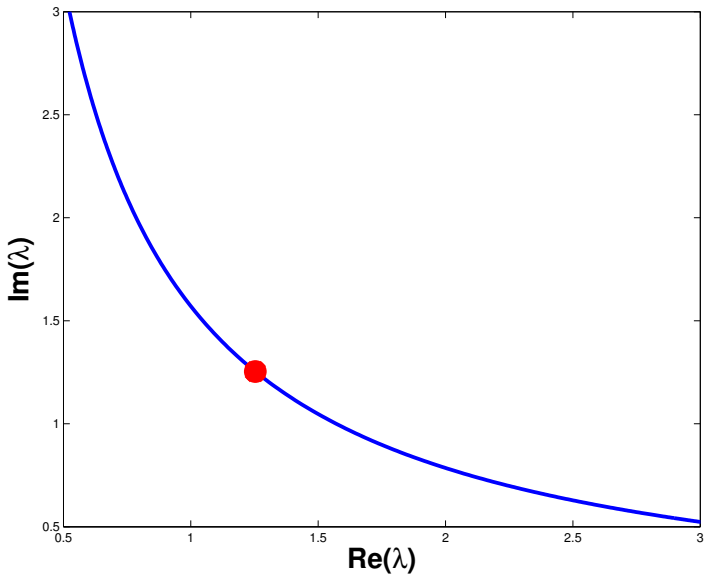

Figure 2-11: *Plot of $\lambda(\xi, k, \epsilon)$ for $k = \epsilon = \pi$ and $\xi \in [1, 2k]$. The circle represents where $\lambda_R = \lambda_I$, which is exactly at $\sqrt{\frac{\epsilon}{2}}$.*

Remark 2.12:

$$\lambda_R(k, k, \epsilon) = \lambda_I(k, k, \epsilon) = \sqrt{\frac{\epsilon}{2}}. \tag{2.37}$$

The following Proposition was used earlier in the chapter but we give a more detailed proof here.

Proposition 2.13:
If $\xi \in \mathbb{R}$ then,

$$\lambda_R(\xi, k, \epsilon) > 0 \text{ and } \lambda_I(\xi, k, \epsilon) > 0, \text{ for all } \epsilon, k \in \mathbb{R}_+.$$

Proof. Since $\lambda(\xi, k, \epsilon)$ is in the first quadrant $\lambda_R(\xi, k, \epsilon) \geq 0$ and $\lambda_I(\xi, k, \epsilon) \geq 0$. Now if we recall (2.28), (2.29) then,

$$\xi^2 - k^2 + i\epsilon = (\lambda_R(\xi, k, \epsilon) + i\lambda_I(\xi, k, \epsilon))^2 .$$

If either $\lambda_R(\xi, k, \epsilon) = 0$ or $\lambda_I(\xi, k, \epsilon) = 0$, for some choice of ξ, then the right hand side of the above equation is purely real which is a contradiction as we choose $\epsilon > 0$. So $\lambda_R(\xi, k, \epsilon) > 0$ and $\lambda_I(\xi, k, \epsilon) > 0$, for all $\xi \in \mathbb{R}$. □

Lemma 2.14:
The functions $\lambda_R(\xi, k, \epsilon)$ and $\lambda_I(\xi, k, \epsilon)$ satisfy the following ODEs,

$$\lambda_R(\xi, k, \epsilon)\lambda_R'(\xi, k, \epsilon) - \lambda_I(\xi, k, \epsilon)\lambda_I'(\xi, k, \epsilon) = \xi, \tag{2.38}$$

$$\lambda_R(\xi, k, \epsilon)\lambda_I'(\xi, k, \epsilon) + \lambda_I(\xi, k, \epsilon)\lambda_R'(\xi, k, \epsilon) = 0. \tag{2.39}$$

where $'$ denotes the derivative with respect to ξ.

Proof. Recall that $\lambda^2(\xi, k, \epsilon) = \xi^2 - k^2 + i\epsilon$. Taking the first derivative with respect to ξ gives,

$$\begin{aligned} 2\lambda(\xi, k, \epsilon)\lambda'(\xi, k, \epsilon) &= 2\xi, \\ \text{So, } \lambda(\xi, k, \epsilon)\lambda'(\xi, k, \epsilon) &= \xi. \end{aligned} \tag{2.40}$$

Now using the fact that $\lambda(\xi, k, \epsilon) = \lambda_R(\xi, k, \epsilon) + i\lambda_I(\xi, k, \epsilon)$ we obtain,

$$\begin{aligned} \xi &= (\lambda_R(\xi, k, \epsilon) + i\lambda_I(\xi, k, \epsilon)) \left(\lambda_R'(\xi, k, \epsilon) + i\lambda_I'(\xi, k, \epsilon)\right), \\ &= \lambda_R(\xi, k, \epsilon)\lambda_R'(\xi, k, \epsilon) - \lambda_I(\xi, k, \epsilon)\lambda_I'(\xi, k, \epsilon) \\ &\quad + i\lambda_R(\xi, k, \epsilon)\lambda_I'(\xi, k, \epsilon) + i\lambda_I(\xi, k, \epsilon)\lambda_R'(\xi, k, \epsilon). \end{aligned} \tag{2.41}$$

Therefore taking real and imaginary parts of (2.41) gives one the desired result. □

Lemma 2.15:
For all $\xi \in \mathbb{R} \setminus \{0\}$, $\lambda_R'(\xi, k, \epsilon) \neq 0$ and $\lambda_I'(\xi, k, \epsilon) \neq 0$.

Proof. The ODE (2.39) together with Proposition 2.13 shows us that if $\lambda'_R(\xi,k,\epsilon)=0$ then $\lambda'_I(\xi,k,\epsilon)=0$. Similarly if $\lambda'_I(\xi,k,\epsilon)=0$ then $\lambda'_R(\xi,k,\epsilon)=0$. However (2.38) tells us that $\lambda'_R(\xi,k,\epsilon)=\lambda'_I(\xi,k,\epsilon)=0$ only if $\xi=0$. Hence the result follows. □

Lemma 2.16:
For all $\xi\in\mathbb{R}\setminus\{0\}$, $\operatorname{sgn}(\lambda'_R(\xi,k,\epsilon))=-\operatorname{sgn}(\lambda'_I(\xi,k,\epsilon))$. *Where the sign function* $\operatorname{sgn}(x)$ *is defined as the following,*

$$\operatorname{sgn}(x):=\begin{cases}-1 & \text{if } x<0,\\ 0 & \text{if } x=0,\\ 1 & \text{if } x>0.\end{cases}$$

Proof. From (2.39) we have,

$$\lambda_R(\xi,k,\epsilon)\lambda'_I(\xi,k,\epsilon)=-\lambda_I(\xi,k,\epsilon)\lambda'_R(\xi,k,\epsilon).$$

The result then follows by Proposition 2.13. □

Lemma 2.17:
The only solution of $\lambda'_R(\xi,k,\epsilon)=-\lambda'_I(\xi,k,\epsilon)$ *when* $\xi\neq 0$ *is* $\xi=k$.

Proof. Suppose that for some ξ,

$$\lambda'_R\left(\xi,k,\epsilon\right)=-\lambda'_I\left(\xi,k,\epsilon\right).$$

Then if we multiply the above by $\lambda_R\left(\xi,k,\epsilon\right)$ and use (2.39) we have that,

$$\begin{aligned}\lambda_R\left(\xi,k,\epsilon\right)\lambda'_R\left(\xi,k,\epsilon\right)&=-\lambda_R\left(\xi,k,\epsilon\right)\lambda'_I\left(\xi,k,\epsilon\right),\\ &=\lambda_I\left(\xi,k,\epsilon\right)\lambda'_R\left(\xi,k,\epsilon\right).\end{aligned}$$

Then combining this with Lemma 2.15 we have that $\lambda_R\left(\xi,k,\epsilon\right)=\lambda_I\left(\xi,k,\epsilon\right)$ which has only one solution at $\xi=k$. □

We now prove some useful results about the behaviour of $\lambda_R(\xi,k,\epsilon)$ and $\lambda_I(\xi,k,\epsilon)$ as $k\to\infty$, evaluated at the minimum and maximum allowable frequencies $\xi=\xi_{min}$ and ξ_{max}. But before doing so the following remark explains how these minimum and maximum frequencies are chosen.

Remark 2.18 (Relevant range of ξ to be considered)**:**
Throughout this thesis we consider that ξ lies in the following range $\xi_{min} \leq \xi \leq \xi_{max}$ where,

$$\xi_{min} \geq 0 \quad \textit{and is a constant independent of } k$$
$$\xi_{max} = \frac{\pi}{h}.$$

In general we consider $h = \frac{\pi}{\eta k}$ and hence

$$\xi_{max} = \eta k. \quad \textit{where} \quad \eta > \sqrt{2}.$$

If one considers a discretisation of 10 grid points then $h = \frac{\pi}{5k}$ and hence

$$\xi_{max} = 5k.$$

As wavelength is given by $\Lambda := \frac{2\pi}{k}$ and we consider a grid spacing given by $h = \frac{\pi}{5k}$ then it follows that $\Lambda = 10h$. Therefore one says that 10 grid points per wavelength are achieved given this mesh spacing. To give an explanation for the choices of ξ_{min} and ξ_{max} given we consider a model problem on the unit line which is discretised with a grid with equidistant spacing. If we are solving the Helmholtz equation (2.11) on this unit line then we expect our solutions to be oscillatory like a sine wave. The solution with minimum frequency is that which joins the end points of the interval, given by the blue line in Figure 2-12. This would have a corresponding wavelength $\Lambda_{min} = 2$, and hence a frequency of $\xi_{min} := \frac{2\pi}{\Lambda_{min}} = \pi$. The corresponding solution with maximum frequency is that which oscillates passing through each grid point as shown by the red line in Figure 2-12. The wavelength in this case would be $\Lambda_{max} = 2h$, and hence a frequency of $\xi_{max} := \frac{2\pi}{\Lambda_{max}} = \frac{\pi}{h}$. Hence the choices which we state above.

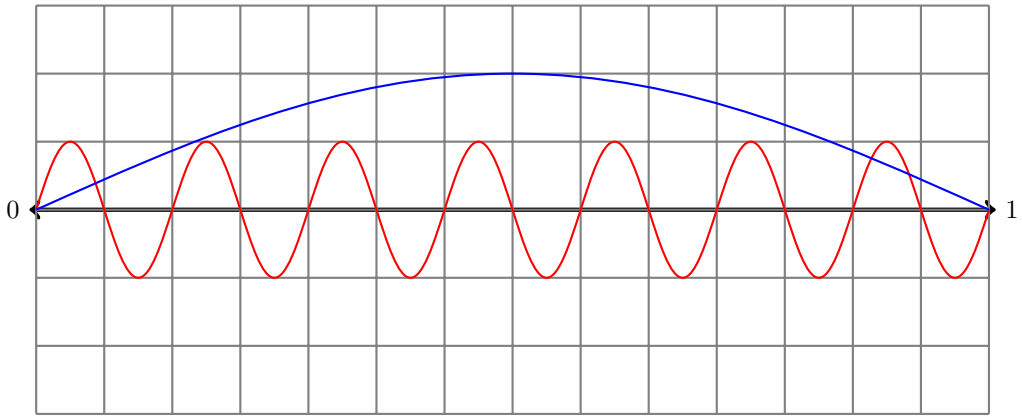

Figure 2-12: *Cartoon of waves with minimum (blue) and maximum (red) allowable frequencies.*

Lemma 2.19:
Assuming that $\epsilon = k^\delta$, $\delta \in [0,2]$, *and* $\xi_{min} \geq 0$,

$$\begin{aligned}\lambda_R(\xi_{min}, k, \epsilon) &= \frac{k^{\delta-1}}{2} + \mathcal{O}\left(k^{\delta-3}\right), \\ \lambda_I(\xi_{min}, k, \epsilon) &= k + \mathcal{O}\left(k^{-1}\right).\end{aligned} \tag{2.42}$$

Proof. If we recall (2.28) and (2.29) and evaluate at $\xi = \xi_{min}$,

$$\begin{aligned}\lambda_R(\xi_{min}, k, \epsilon) + i\lambda_I(\xi_{min}, k, \epsilon) &= \sqrt{\xi_{min}^2 - k^2 + i\epsilon}, \\ &= ik\sqrt{1 - \frac{\xi_{min}^2}{k^2} - i\frac{\epsilon}{k^2}}.\end{aligned}$$

If we then recall that,

$$\sqrt{1-z} = 1 - \frac{z}{2} - \frac{z^2}{8} - \frac{z^3}{16} - ..., \text{ for sufficiently small complex } z.$$

Then it follows that,

$$\begin{aligned}\lambda_R(\xi_{min}, k, \epsilon) + i\lambda_I(\xi_{min}, k, \epsilon) &= ik\left(1 - \frac{1}{2}\left(\frac{\xi_{min}^2}{k^2} + i\frac{\epsilon}{k^2}\right)\right) + \mathcal{O}(\epsilon k^{-3}), \\ &= \frac{\epsilon}{2k} + i\left(k - \frac{\xi_{min}^2}{2k}\right) + \mathcal{O}(\epsilon k^{-3}).\end{aligned}$$

Then if we substitute $\epsilon = k^\delta$ the result follows. □

Lemma 2.20:
Assuming that $\epsilon = k^\delta$, $\delta \in [0,2]$, *and* $\xi_{max} = \eta k$,

$$\begin{aligned}\lambda_R(\xi_{max}, k, \epsilon) &= \gamma k\left(1 + \mathcal{O}\left(k^{2(\delta-2)}\right)\right), \\ \lambda_I(\xi_{max}, k, \epsilon) &= \frac{k^{\delta-1}}{2\gamma}\left(1 + \mathcal{O}\left(k^{2(\delta-2)}\right)\right), \text{ where } \gamma = \sqrt{\eta^2 - 1} > 1.\end{aligned} \tag{2.43}$$

Proof. Once again we write,

$$\lambda_R(\xi_{max}, k, \epsilon) + i\lambda_I(\xi_{max}, k, \epsilon) = \sqrt{\xi_{max}^2 - k^2 + i\epsilon}.$$

Recalling that $\xi_{max} = \eta k$, it follows that

$$\begin{aligned}\lambda_R(\xi_{max}, k, \epsilon) + i\lambda_I(\xi_{max}, k, \epsilon) &= \sqrt{(\eta^2-1)k^2 + i\epsilon}, \\ &= \gamma k\sqrt{1 + \frac{i\epsilon}{\gamma^2 k^2}}, \text{ where } \gamma = \sqrt{(\eta^2-1)} > 1.\end{aligned}$$

If we then Taylor expand the square root for large k it then follows that,

$$\begin{aligned}\lambda_R(\xi_{max},k,\epsilon)+i\lambda_I(\xi_{max},k,\epsilon) &= \gamma k\left(1+\frac{i\epsilon}{2\gamma^2k^2}+\frac{\epsilon^2}{8\gamma^4k^4}-\frac{i\epsilon^3}{16\gamma^6k^6}+\mathcal{O}\left(\frac{\epsilon^4}{k^8}\right)\right),\\ &= \gamma k\left(1+\frac{\epsilon^2}{8\gamma^4k^4}\right)+\frac{i\epsilon}{2\gamma k}\left(1-\frac{\epsilon^2}{8\gamma^4k^4}\right)+\mathcal{O}\left(\frac{\epsilon^4}{k^7}\right).\end{aligned}$$

Then if we substitute $\epsilon = k^\delta$ the result follows. □

2.3.4 Comparison with the classical Schwarz algorithm

As mentioned at the start of this section it is expected that the optimised Schwarz algorithm (2.17) with a good choice of σ should perform better than the classical Schwarz algorithm which we show below,

$$\begin{aligned}\left(-\Delta-k^2+i\epsilon\right)u_1^n(x,y) &= f(x,y),\ \text{in } \Omega_1\\ u_1^n(x,y) &= u_2^{n-1}(x,y),\ \text{on } \Gamma_1\\ \left(-\Delta-k^2+i\epsilon\right)u_2^{n+1}(x,y) &= f(x,y),\ \text{in } \Omega_2\\ u_2^{n+1}(x,y) &= u_1^n(x,y),\ \text{on } \Gamma_2.\end{aligned}\tag{2.44}$$

If we perform the same Fourier analysis that we did for the optimised Schwarz algorithm (2.17) then we can derive a similar expression for the convergence rate of the classical algorithm (2.44). The result of this is given in the following Theorem. We make the reader aware that the following results in this subsection are all original.

Theorem 2.21:
The convergence rate of the Fourier transform of the classical Schwarz algorithm (2.44) *is given by,*

$$\rho^C(\xi,k,\epsilon,L) = e^{-2\lambda(\xi,k,\epsilon)L} < 1,\ \forall\ \xi\in\mathbb{R}_+,\ \textit{and}\ L>0.\tag{2.45}$$

Thus if there is no overlap ($L=0$) then,

$$\rho^C(\xi,k,\epsilon,0) = 1,\ \forall\ \xi\in\mathbb{R}_+.\tag{2.46}$$

Proof. We prove (2.45) by following the same steps as those for the proof of Theorem 2.10. We omit these details for brevity. □

As we can see from from (2.46), if there is no overlap present then the classical Schwarz algorithm will not converge. Therefore we are already at an advantage using the optimised method as we do not require overlap for the algorithm to converge as the convergence rate (2.22) will always be less than 1, which shall be proven later.

A result of interest is how the convergence rate (2.45) for the classical method behaves as k increases. We start by showing that (2.45) has only one maximum in ξ, and then show how (2.45) behaves for increasing k at this value of ξ.

Corollary 2.22:
The convergence rate $\left|\rho^C(\xi,k,\epsilon,L)\right|$ *attains its maximum when* $\xi = 0$.

Proof. We prove this simply by taking a first derivative of (2.45) in ξ,

$$\begin{aligned}\frac{\partial}{\partial\xi}\left|\rho^C(\xi,k,\epsilon,L)\right| &= \frac{\partial}{\partial\xi}e^{-2\lambda_R(\xi,k,\epsilon)L},\\ &= -2Le^{-2\lambda_R(\xi,k,\epsilon)L}\left(\frac{\partial}{\partial\xi}\lambda_R\left(\xi,k,\epsilon\right)\right)\end{aligned} \tag{2.47}$$

If we recall that $\lambda\left(\xi,k,\epsilon\right) = \sqrt{\xi^2 - k^2 + i\epsilon}$ then it is simple to show that,

$$\begin{aligned}\frac{\partial}{\partial\xi}\lambda\left(\xi,k,\epsilon\right) &= \frac{\xi}{\lambda\left(\xi,k,\epsilon\right)},\\ &= \frac{\xi}{\lambda_R\left(\xi,k,\epsilon\right) + i\lambda_I\left(\xi,k,\epsilon\right)},\\ &= \xi\left(\frac{\lambda_R\left(\xi,k,\epsilon\right)}{\lambda_R^2\left(\xi,k,\epsilon\right) + \lambda_I^2\left(\xi,k,\epsilon\right)} - i\frac{\lambda_I\left(\xi,k,\epsilon\right)}{\lambda_R^2\left(\xi,k,\epsilon\right) + \lambda_I^2\left(\xi,k,\epsilon\right)}\right).\end{aligned}$$

Hence,

$$\frac{\partial}{\partial\xi}\lambda_R\left(\xi,k,\epsilon\right) = \xi\frac{\lambda_R\left(\xi,k,\epsilon\right)}{\lambda_R^2\left(\xi,k,\epsilon\right) + \lambda_I^2\left(\xi,k,\epsilon\right)}$$

If we then insert this into (2.47) it follows that,

$$\frac{\partial}{\partial\xi}\left|\rho^C(\xi,k,\epsilon,L)\right| = -2L\xi\frac{\lambda_R\left(\xi,k,\epsilon\right)}{\lambda_R^2\left(\xi,k,\epsilon\right) + \lambda_I^2\left(\xi,k,\epsilon\right)}e^{-2\lambda_R(\xi,k,\epsilon)L}. \tag{2.48}$$

Clearly (2.48) is zero if and only if $\xi = 0$. Therefore $\xi = 0$ is the only critical point of (2.45). Then we can observe that for all $\xi > 0$ (2.48) is decreasing. Hence (2.45) attains its maximum when $\xi = 0$. □

Corollary 2.23:
The overlapping classical Schwarz algorithm (2.44) *with* $L = C_L h$ *and* $\epsilon = k^\delta$ *(where* $\delta = [0,2]$*) has an asymptotic convergence rate* (2.45) *given by,*

$$\max_{\xi\in\mathbb{R}}\left|\rho^C(\xi,k,\epsilon,L)\right| = 1 - \frac{C_L\pi}{\eta}k^{\delta-2} + \mathcal{O}\left(k^{2(\delta-2)}\right),\ \textit{where}\ \gamma = \sqrt{\eta^2 - 1}. \tag{2.49}$$

Proof. As we have proven previously, at $\xi = 0$ (2.45) attains its maximum value and

hence,

$$\begin{aligned}\max_{\xi\in\mathbb{R}}\left|\rho^C(\xi,k,\epsilon,C_Lh)\right| &= \left|e^{-2\lambda(0,k,\epsilon)C_Lh}\right|,\\ &= e^{-2\lambda_R(0,k,\epsilon)C_Lh}.\end{aligned} \tag{2.50}$$

Then assuming that $h=\frac{\pi}{\eta k}$ our previous equation (2.50) gives

$$\max_{\xi\in\mathbb{R}}\left|\rho^C(\xi,k,\epsilon,C_Lh)\right| = e^{-\frac{2C_L\pi}{\eta}\frac{\lambda_R(0,k,\epsilon)}{k}}.$$

If we recall (2.42) which is the leading order approximation of $\lambda_R(0,k,\epsilon)$ then it follows that,

$$\max_{\xi_{min}\leq\xi\leq\xi_{max}}\left|\rho^C(\xi,k,\epsilon,C_Lh)\right| = e^{-\frac{C_L\pi}{\eta}k^{\delta-2}}. \tag{2.51}$$

We can then perform a Taylor expansion of (2.51) in powers of $\frac{1}{k}$ for k large enough. As $\exp(-k^{\delta-2})=\sum_{j=0}^{\infty}\frac{\left(-k^{\delta-2}\right)^j}{j!}$ it then follows that,

$$\max_{\xi_{min}\leq\xi\leq\xi_{max}}\left|\rho^C(\xi,k,\epsilon,C_Lh)\right| = 1-\frac{C_L\pi}{\eta}k^{\delta-2}+\mathcal{O}\left(k^{2(\delta-1)}\right). \tag{2.52}$$

□

Therefore if ϵ is chosen such that $\epsilon=k^2$, $\delta=2$ and Corollary 2.23 says that we should expect a convergence rate which does not depend asymptotically on k. In Figure 2-13 the convergence rate (2.45) is plotted for increasing ϵ and fixed $k=5$, $h=\frac{\pi}{5k}$ and $L=h$. We can see from this Figure that even for relatively low ϵ the convergence rate (2.45) is not near to one and damps the high frequency ξ effectively. If one increases the absorption such that $\epsilon\sim k^{\frac{3}{2}}$ or $\epsilon\sim k^2$, then the lower frequencies are effectively damped too. And indeed when $\epsilon\sim k^2$ then the classical Schwarz algorithm (2.44) will converge in a constant number of iterations for increasing k. This is useful as we could use one (or several) iterations of (2.44) as the preconditioner for GMRES in (2.14). However a choice of $\epsilon\sim k^2$ may not be best for the preconditioned solver, even though it best for solving (2.44). In Figure 2-14 we examine the result of Corollary 2.23. As the maximum of (2.45) is attained at $\xi=0$ we therefore plot (2.45) for increasing k with a fixed ϵ and overlap L. What we observe is that for increasing k the convergence rate becomes closer to 1. The maximum value of the convergence rate increases as k increases which agrees with the asymptotic result in (2.49).

In the following chapters we shall look at improving on the classical result so that we have an algorithm which does not require overlap to converge, and for which the convergence rate does not deteriorate so quickly as k increases.

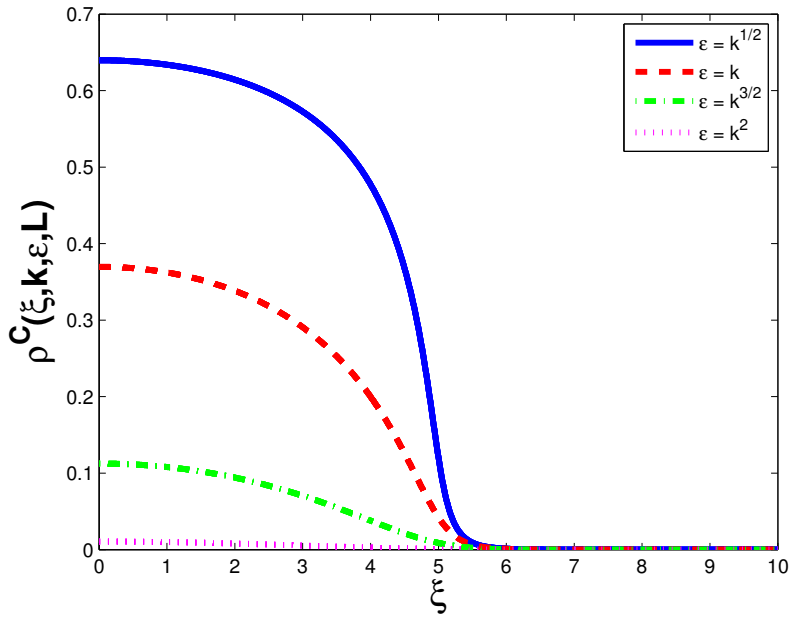

Figure 2-13: *Plot of the convergence rate* $\left|\rho^C(\xi,k,\epsilon,L)\right|$ *as a function of* ξ *for different* ϵ. *Here we use* $k=5$, $h=\frac{\pi}{5k}$ *and an overlap of* $L=h$.

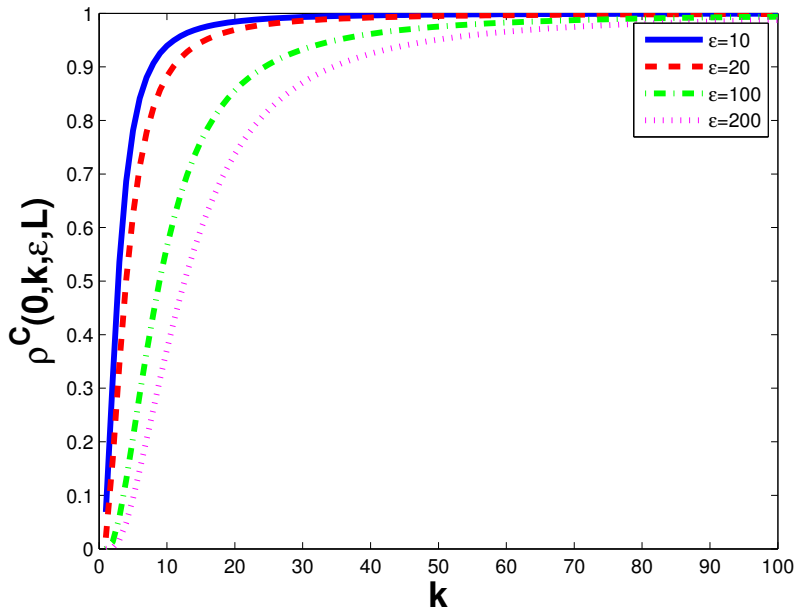

Figure 2-14: *Plot of* $\max_\xi \left|\rho^C(\xi,k,\epsilon,L)\right|$ *as a function of* k. *Here we increase* ϵ *and fix* $h=\frac{\pi}{5k}$ *and an overlap of* $L=h$.

2.4 How to choose $\sigma(\xi)$ to make the Schwarz algorithm converge faster

We now return to the Schwarz algorithm with Robin condition (2.17), with convergence rate given by Theorem 2.10. Our aim is to choose a symbol σ so that the resulting operator S in (2.20) is easy to use but also has a good convergence rate ρ in (2.22).

A possible approach is to choose σ as a low order approximation of $\lambda(\xi, k, \epsilon)$ with respect to ξ.

2.4.1 Approximation of $\lambda(\xi, k, \epsilon)$ by Taylor expansion

As our first approximation we try using the lower order terms from the Taylor expansion of $\lambda(\xi, k, \epsilon)$ for small ξ. This idea is nothing new and goes back at least to Engquist and Majda [18]. However we do not know of a reference for the discussion of this in the case that $\epsilon \neq 0$. The Taylor expansion of $\lambda(\xi, k, \epsilon)$ about $\xi = 0$ is given by,

$$\begin{aligned}\lambda(\xi, k, \epsilon) &= \sqrt{\xi^2 - k^2 + i\epsilon}, \\ &= \lambda(0, k, \epsilon) + \frac{\partial \lambda(0, k, \epsilon)}{\partial \xi}(\xi - 0) + ..., \\ &= \lambda(0, k, \epsilon) + \frac{\xi^2}{2\lambda(0, k, \epsilon)} + \mathcal{O}(\xi^4). \end{aligned} \tag{2.53}$$

So a plausible choice for σ would be to take the first few terms in the Taylor expansion of $\lambda(\xi, k, \epsilon)$, e.g.

$$\sigma = \sqrt{-k^2 + i\epsilon} + \frac{\xi^2}{2\sqrt{-k^2 + i\epsilon}}. \tag{2.54}$$

Ultimately we will need the action of the operator S in order to implement (2.17) then we must take the inverse Fourier transform of (2.54). This is

$$\begin{aligned}(S\phi)(y) &= \mathcal{F}^{-1}\left\{\left(\sqrt{-k^2 + i\epsilon} + \frac{\xi^2}{2\sqrt{-k^2 + i\epsilon}}\right)\hat{\phi}(\xi,)\right\} \\ &= \sqrt{-k^2 + i\epsilon}\mathcal{F}^{-1}\left\{\hat{\phi}(\xi)\right\} + \frac{1}{2\sqrt{-k^2 + i\epsilon}}\mathcal{F}^{-1}\left\{\xi^2\hat{\phi}(\xi)\right\}, \\ &= \left(\sqrt{-k^2 + i\epsilon} - \frac{1}{2\sqrt{-k^2 + i\epsilon}}\partial_{yy}^2\right)\phi(y) =: S_2^T\phi(y). \end{aligned} \tag{2.55}$$

This is called the Taylor order two approximation denoted S_2^T. A cruder approximation would be to use the first term in (2.55). This would lead to so called *zeroth* order Taylor

approximation of the form

$$S_0^T = \lambda(0,k,\epsilon) = \sqrt{-k^2 + i\epsilon}. \tag{2.56}$$

The convergence rate (2.22) for S_0^T is

$$\rho_0^T(\xi,k,\epsilon,\sigma,L) = \left(\frac{-\lambda(\xi,k,\epsilon)+\lambda(0,k,\epsilon)}{\lambda(\xi,k,\epsilon)+\lambda(0,k,\epsilon)}\right)^2 e^{-2\lambda(\xi,k,\epsilon)L}.$$

Then if we use (2.29) this gives,

$$\left|\rho_0^T(\xi,k,\epsilon,\sigma,L)\right| = \frac{\left(\lambda_R(0,k,\epsilon)-\lambda_R(\xi,k,\epsilon)\right)^2+\left(\lambda_I(0,k,\epsilon)-\lambda_I(\xi,k,\epsilon)\right)^2}{\left(\lambda_R(0,k,\epsilon)+\lambda_R(\xi,k,\epsilon)\right)^2+\left(\lambda_I(0,k,\epsilon)+\lambda_I(\xi,k,\epsilon)\right)^2} e^{-2\lambda_R(\xi,k,\epsilon)L}, \tag{2.57}$$

We now prove that if we use this low order Taylor choice of transmission condition (2.56) then the Schwarz algorithm (2.17) converges even if we use no overlap, $L = 0$. We obtain a similar result for higher order conditions. Again we make the reader aware that all the results which follow in this section are original.

Theorem 2.24:
The convergence rate (2.57) *of the Schwarz algorithm* (2.17) *with* $S = S_0^T$ *satisfies*

$$\left|\rho_0^T(\xi,k,\epsilon,\sigma,L)\right| < e^{-2L\lambda_R(\xi,k,\epsilon)}, \text{ for all } \xi \in \mathbb{R}, \text{ and } L \geq 0.$$

Proof. We firstly recall the convergence rate with Taylor transmission conditions (2.57) with overlap $L = 0$,

$$\left|\rho_0^T(\xi,k,\epsilon,\sigma,0)\right| = \frac{\left(\lambda_R(0,k,\epsilon)-\lambda_R(\xi,k,\epsilon)\right)^2+\left(\lambda_I(0,k,\epsilon)-\lambda_I(\xi,k,\epsilon)\right)^2}{\left(\lambda_R(0,k,\epsilon)+\lambda_R(\xi,k,\epsilon)\right)^2+\left(\lambda_I(0,k,\epsilon)+\lambda_I(\xi,k,\epsilon)\right)^2}.$$

Then if we recall from (2.30) that $\lambda_R, \lambda_I > 0$ it follows that,

$$\left|\rho_0^T(\xi,k,\epsilon,\sigma,0)\right| < 1.$$

If we were to include an overlap L then if we recall (2.57) and the above results then,

$$\begin{aligned}
\left|\rho_0^T(\xi,k,\epsilon,\sigma,L)\right| &< \left|e^{-2L\lambda(\xi,k,\epsilon)}\right|, \\
&= \left|e^{-2L(\lambda_R(\xi,k,\epsilon)-i\lambda_I(\xi,k,\epsilon))}\right|, \text{ recalling (2.29)}, \\
&= \left|e^{-2L\lambda_R(\xi,k,\epsilon)}\right|\left|e^{-i2L\lambda_I(\xi,k,\epsilon)}\right|.
\end{aligned}$$

Then as $\left|e^{-i2L\lambda_I(\xi,k,\epsilon)}\right| = 1$ this gives,

$$\left|\rho_0^T(\xi,k,\epsilon,\sigma,L)\right| < e^{-2L\lambda_R(\xi,k,\epsilon)},$$

and as $\lambda_R > 0$ then we can see that for $L > 0$ the convergence rate will decay exponentially when $\xi \to \infty$. This can clearly be seen in Figures 2-16 and 2-18. □

Figures 2-15, 2-16, 2-17 and 2-18 show the convergence rate (2.57) plotted against ξ with fixed k, h and a choice of either (2.56) or (2.55) with no overlap or an overlap of $L = h$. In these plots the convergence rate for different values of ϵ is plotted. One can see immediately that the convergence rate (2.57) for both the non overlapping and overlapping choices is indeed less than 1 for all frequencies ξ, different choices of ϵ and interface condition. Also one can see that, for the non overlapping method, as we are using a low frequency approximation for the transmission conditions we achieve better convergence for lower frequencies, with the convergence deteriorating at high frequencies.

It is worth pointing out that ρ_0^T never actually reaches 1 in the plot in Figures 2-15 and 2-17. In a numerical implementation (on a uniform grid for example) there will exist a maximum allowable frequency. In the case of a uniform grid with mesh spacing h this can be found (see Remark 2.18) to be $\xi_{max} = \eta k$, for some constant we choose $\eta > \sqrt{2}$. We remind the reader that we choose $h = \frac{\pi}{\eta}k^{-1}$. All of the following results could be repeated for $h \sim k^{-\alpha}$, where $1 < \alpha \leq 2$, to further abate the *pollution effect* [25].

It is useful to know when (2.57) attains its maximum with respect to ξ, especially for the asymptotic expressions for (2.57) which will follow. Figures 2-15 (zeroth order (2.56)) and 2-17 (second order (2.55)) show the convergence rate (2.57) with no overlap plotted for increasing ξ with fixed k and ϵ. clearly show that (2.57) attains its maximum for a non overlapping Schwarz method with Taylor conditions when $\xi = \xi_{max}$.

We now turn our attention to the case when (2.57) has an overlap parameter $L > 0$. Figures 2-16 and 2-18 plot (2.57) for increasing ξ with a fixed k, ϵ and overlap L. What we can observe from these Figures is that there is an obvious maximum of (2.57) which is situated near to $\xi = k$. We present these numerical findings in the following two conjectures. These are not proved but simply arise from our observations.

Conjecture 2.25:
The convergence rate of the non overlapping Schwarz algorithm (2.17) *with zeroth order Taylor transmission conditions* (2.53) *attains its maximum at* $\xi = \xi_{max}$.

Conjecture 2.26:
The convergence rate of the overlapping Schwarz algorithm (2.17) *with zeroth order*

Taylor transmission conditions (2.53) *attains its maximum near to* $\xi = k$.

We now use Conjecture 2.25 and 2.26 to prove the following results about the asymptotic behaviour of the non overlapping and overlapping Schwarz algorithm (2.17) for increasing k.

Theorem 2.27:
The non overlapping Schwarz algorithm (2.17) *with zeroth order Taylor transmission conditions* (2.53) *with* $\xi_{max} = \eta k$ *($\eta > sqrt2$), and* $\epsilon = k^{\delta}$ *(where* $\delta = [0, 2]$*) has an asymptotic convergence rate* (2.57) *given by,*

$$\max_{\xi_{min} \leq \xi \leq \xi_{max}} \left| \rho_0^T \left(\xi, k, \epsilon, \sigma, 0 \right) \right| = 1 - \frac{2}{\gamma} k^{\delta - 2} + \mathcal{O}\left(k^{2(\delta-2)} \right), \text{ where } \gamma = \sqrt{\eta^2 - 1}. \quad (2.58)$$

as $k \to \infty$.

Proof. By Conjecture 2.25 we expect the convergence convergence of the non overlapping method to be nearest to one at the highest frequency ξ_{max}. Therefore we choose to evaluate the convergence rate (2.57) at $\xi = \xi_{max} = \eta k$,

$$\left| \rho_0^T \left(\eta k, k, \epsilon, \sigma, 0 \right) \right| = \frac{(\lambda_R(0, k, \epsilon) - \lambda_R(\eta k, k, \epsilon))^2 + (\lambda_I(0, k, \epsilon) - \lambda_I(\eta k, k, \epsilon)))^2}{(\lambda_R(0, k, \epsilon) + \lambda_R(\eta k, k, \epsilon))^2 + (\lambda_I(0, k, \epsilon) + \lambda_I(\eta k, k, \epsilon))^2}.$$

We can simplify this further by using some asymptotic results about the behaviour of the real and imaginary parts of λ. If we substitute the results of Lemma 2.19 (with $\xi_{min} = 0$) and Lemma 2.20 the above equation then becomes,

$$\left| \rho_0^T \left(\eta k, k, \epsilon, \sigma, 0 \right) \right| \sim \frac{(\frac{k^{\delta-1}}{2} - \gamma k)^2 + (k - \frac{k^{\delta-1}}{2\gamma})^2}{(\frac{k^{\delta-1}}{2} + \gamma k)^2 + (k + \frac{k^{\delta-1}}{2\gamma})^2}, \quad (2.59)$$

where we have dropped the higher order terms. If we then perform a Taylor expansion of the above for $k \to \infty$ this gives the desired result (2.58). □

Theorem 2.27 tells us that for k increasing and a choice of low order Taylor conditions in our Schwarz algorithm we should observe a convergence which behaves like $1 - \mathcal{O}\left(k^{\delta-2}\right)$. An interesting observation one can make is that if we choose $\epsilon = k^2$, then the convergence rate no longer depends on k asymptotically and is constant. This tells us that for this choice of the parameter ϵ we should observe convergence of the Schwarz algorithm (2.17) in a number of iterations which is independent of k. We now prove a similar result to that above for the overlapping Schwarz algorithm.

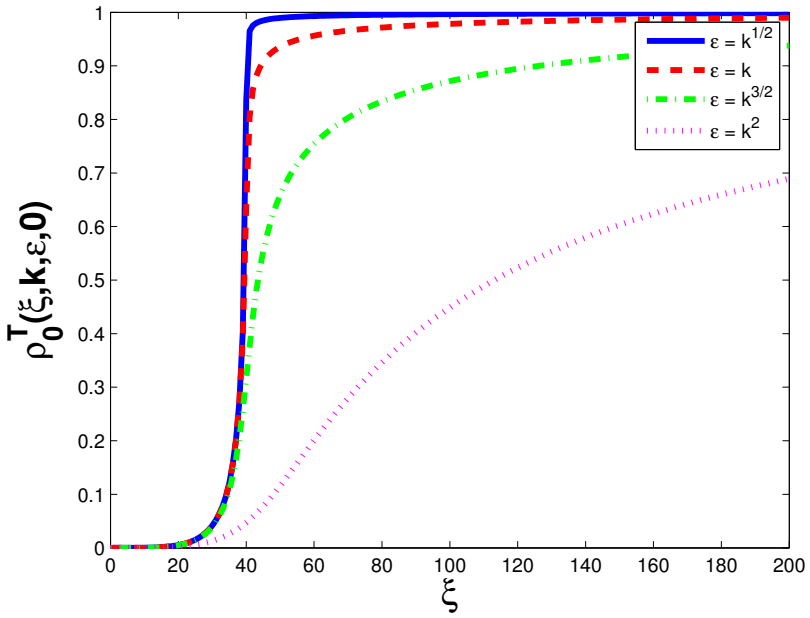

Figure 2-15: *Plot of the convergence rate* $|\rho_0^T(\xi, k, \epsilon, 0)|$ *(no overlap) as a function of* ϵ*. Here we use* $k = 40$ *and* $h = \frac{\pi}{5k}$*.*

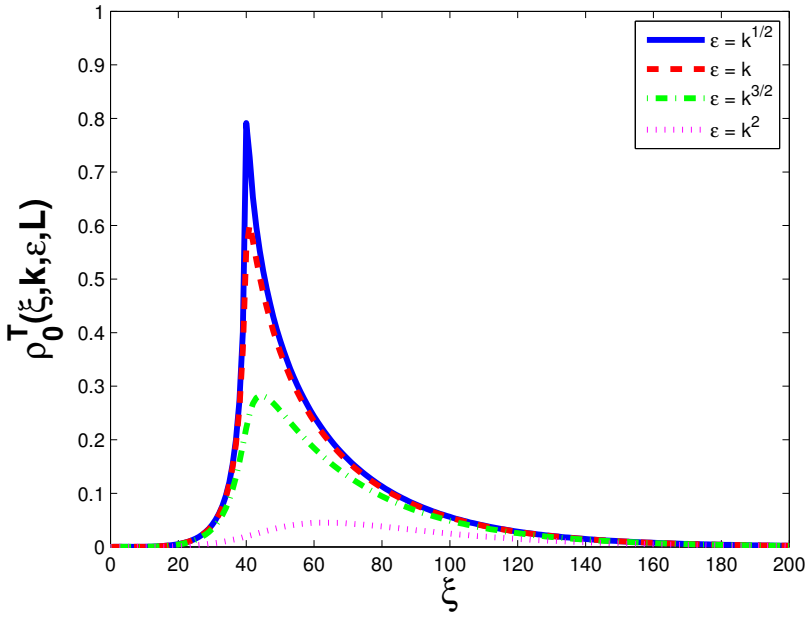

Figure 2-16: *Plot of the convergence rate* $|\rho_0^T(\xi, k, \epsilon, L)|$ *as a function of* ϵ*. Here we use* $k = 40$*,* $h = \frac{\pi}{5k}$ *and an overlap of* $L = h$*.*

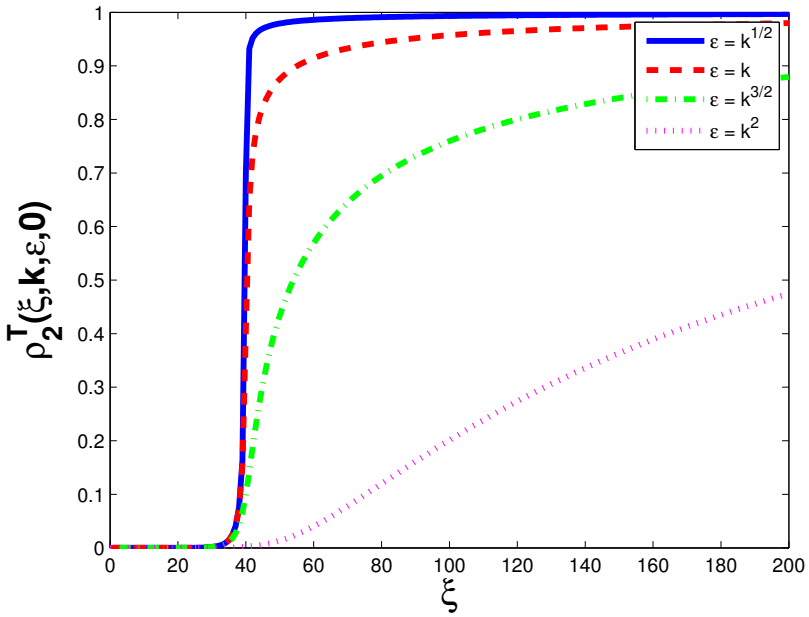

Figure 2-17: *Plot of the convergence rate* $|\rho_2^T(\xi, k, \epsilon, 0)|$ *(no overlap) as a function of* ϵ*. Here we use* $k = 40$ *and* $h = \frac{\pi}{5k}$*.*

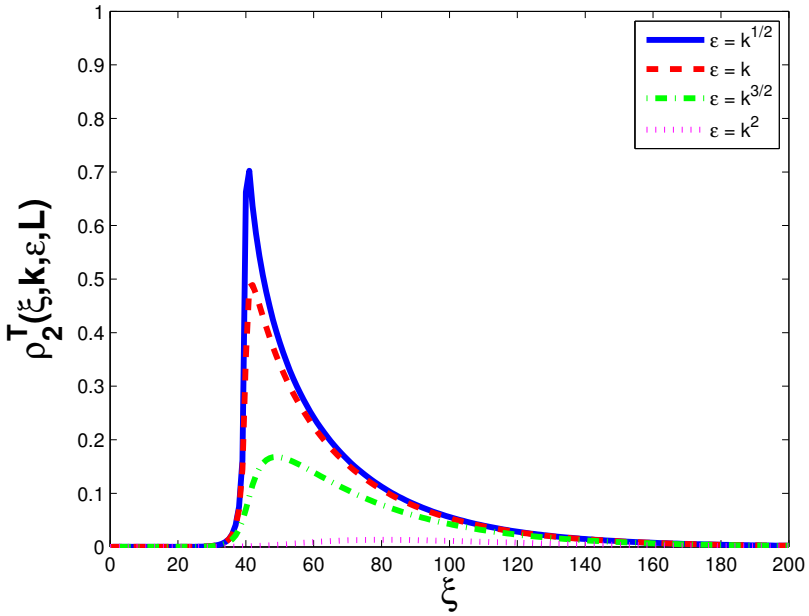

Figure 2-18: *Plot of the convergence rate* $|\rho_2^T(\xi, k, \epsilon, L)|$ *as a function of* ϵ*. Here we use* $k = 40$*,* $h = \frac{\pi}{5k}$ *and an overlap of* $L = h$*.*

Theorem 2.28:
The overlapping Schwarz algorithm (2.17) *with zeroth order Taylor transmission conditions* (2.56)*,* $\xi_{max} = \eta k$*, an overlap of* $L = C_L h$ *where* $C_L \in \mathbb{Z}$*, and* $\epsilon = k^\delta$ *(where* $\delta = [0, 2]$*) has an asymptotic convergence rate* (2.57) *given by,*

$$\max_{\xi_{min} \leq \xi \leq \xi_{max}} \left| \rho_0^T \left(\xi, k, \epsilon, \sigma, C_L h \right) \right| = 1 - \left(2\sqrt{2} + \frac{\sqrt{2}\pi C_L}{\eta} \right) k^{\frac{\delta-2}{2}} + \mathcal{O}\left(k^{\delta-2} \right). \quad (2.60)$$

Proof. By Conjecture 2.26 we know that $\left| \rho_0^T \left(\xi, k, \epsilon, \sigma, C_L h \right) \right|$ has at least one internal maximum and that it occurs when ξ is near to k. Therefore to prove (2.60) we need only evaluate (2.57) at $\xi = k$ and Taylor expand for increasing k. We recall (2.57) and set $\xi = k$ and $L = C_L h$ which gives,

$$\left| \rho_0^T (k, k, \epsilon, \sigma, L) \right| = \frac{\left(\lambda_R(0, k, \epsilon) - \lambda_R(k, k, \epsilon) \right)^2 + \left(\lambda_I(0, k, \epsilon) - \lambda_I(k, k, \epsilon) \right)^2}{\left(\lambda_R(0, k, \epsilon) + \lambda_R(k, k, \epsilon) \right)^2 + \left(\lambda_I(0, k, \epsilon) + \lambda_I(k, k, \epsilon) \right)^2} \left| e^{-2\lambda(k,k,\epsilon) C_L h} \right|.$$

Now using the fact that $\lambda_R = \lambda_I = \sqrt{\frac{\epsilon}{2}}$, and that $\left| e^{-2\lambda(\xi,k,\epsilon) C_L h} \right| = e^{-2\lambda_R(\xi,k,\epsilon) C_L h}$, the above equation simplifies to,

$$\left| \rho_0^T \left(\sigma k, k, \epsilon, \sigma, 0 \right) \right| = \frac{\left(\lambda_R(0, k, \epsilon) - \sqrt{\frac{\epsilon}{2}} \right)^2 + \left(\lambda_I(0, k, \epsilon) - \sqrt{\frac{\epsilon}{2}} \right)^2}{\left(\lambda_R(0, k, \epsilon) + \sqrt{\frac{\epsilon}{2}} \right)^2 + \left(\lambda_I(0, k, \epsilon) + \sqrt{\frac{\epsilon}{2}} \right)^2} e^{-2\lambda_R(\xi,k,\epsilon) C_L h}.$$

As mentioned previously we choose discretisations with a certain number of grid points per wavelength, therefore we fix $h = \frac{\pi}{\eta k}$ and substitute that in to give,

$$\left| \rho_0^T \left(\sigma k, k, \epsilon, \sigma, 0 \right) \right| = \frac{\left(\lambda_R(0, k, \epsilon) - \sqrt{\frac{\epsilon}{2}} \right)^2 + \left(\lambda_I(0, k, \epsilon) - \sqrt{\frac{\epsilon}{2}} \right)^2}{\left(\lambda_R(0, k, \epsilon) + \sqrt{\frac{\epsilon}{2}} \right)^2 + \left(\lambda_I(0, k, \epsilon) + \sqrt{\frac{\epsilon}{2}} \right)^2} e^{-2C_L \frac{\pi}{\eta k} \lambda_R(\xi,k,\epsilon)}.$$

Then by substituting $\epsilon = k^\delta$ and using the expressions for $\lambda_R(0, k, \epsilon)$ and $\lambda_I(0, k, \epsilon)$ (from Lemma 2.19) in the above equation, we can then perform an expansion for $k \to \infty$ to give the required result (2.60). □

This result tells one that with the addition of even a small overlap an improvement in the convergence of $1 - \mathcal{O}\left(k^{\frac{\delta-2}{2}} \right)$ is obtained in comparison to the non overlapping method which is $1 - \mathcal{O}\left(k^{\delta-2} \right)$. Indeed if we compare Figures 2-19 (no overlap) and 2-20 they both plot the convergence rate (2.57), with a zeroth order interface condition (2.56), for increasing k. The difference being that the latter Figure has overlap and the former does not. In Figure 2-19, with no overlap, the convergence rate increases quickly until it almost reaches 1 (but doesn't reach 1). However with the addition of a small amount of overlap in Figure 2-20 the convergence rate is dampened

We now add in the additional second order term, that is choosing $S = S_2^T$ in the hope

that this aids convergence, where S_2^T is as in (2.55). The proofs of theorems 2.29 and 2.30 use the same strategy as those for the *zeroth* order case so we simply state the results.

Theorem 2.29:
The non overlapping Schwarz algorithm (2.17) *with second order Taylor transmission conditions* (2.55) *with* $\xi_{max} = \eta k$, *and* $\epsilon = k^\delta$, *where* $\delta = [0, 2]$, *has an asymptotic convergence rate* (2.57) *given by,*

$$\max_{\xi_{min} \leq \xi \leq \xi_{max}} \left| \rho_2^T \left(\xi, k, \epsilon, \sigma, 0 \right) \right| = 1 - \frac{4\gamma}{\eta^2} k^{\delta - 2} + \mathcal{O}\left(k^{2(\delta-2)} \right), \text{ where } \gamma = \sqrt{\eta^2 - 1}. \tag{2.61}$$

Theorem 2.30:
The overlapping Schwarz algorithm (2.17) *with second order Taylor transmission conditions* (2.55), $\xi_{max} = \sigma k$, *an overlap of* $L = C_L h$ *where* $C_L \in \mathbb{Z}$, *and* $\epsilon = k^\delta$ *(where* $\delta = [0, 2]$*) has an asymptotic convergence rate* (2.57) *given by,*

$$\max_{\xi_{min} \leq \xi \leq \xi_{max}} \left| \rho_2^T \left(\xi, k, \epsilon, \sigma, C_L h \right) \right| = 1 - \left(4\sqrt{2} + \frac{\sqrt{2}\pi C_L}{\eta} \right) k^{\frac{\delta-2}{2}} + \mathcal{O}\left(k^{\delta-2} \right). \tag{2.62}$$

We find that the addition of a second order term from the Taylor expansion of λ makes little difference to the asymptotic behaviour of the convergence rate (2.22). See for example Figures 2-19, 2-21 for a comparison of the convergence factors for a non-overlapping and overlapping method with increasing k. Therefore it is expected that using a second order Taylor approximation when solving with (2.26) will not result in convergence which is much better than simply using a zeroth order Taylor condition. Figures 2-21, 2-22 give a comparison of the convergence factors with and without overlap for increasing k, showing similar results to those with the zeroth order interface condition. If we compare these overlapping results to those of the overlapping classical Schwarz method in Figure 2-14 then it is clear that the convergence rates of methods with Taylor interface conditions do not deteriorate as quickly as k increases. Indeed in Figure 2-14 the convergence rate of the classical algorithm is almost 1 when $k = 20$. In comparison the convergence rates in both Figure 2-20 and 2-22 are both below 0.6 when $k = 20$, and below 0.9 for all k in the range considered. We notice also that the difference of the $\max_\xi |\rho^T|$ between the non-overlapping and overlapping is not terribly pronounced. For example comparing 2-19 and 2-20, when $k = 100$ the largest reduction in $\max_\xi |\rho^T|$ is from 0.4 to 0.3 when $\epsilon = 200$. Otherwise the reduction is much less pronounced. In the next chapter we outline a strategy to further improve the convergence rate of (2.17).

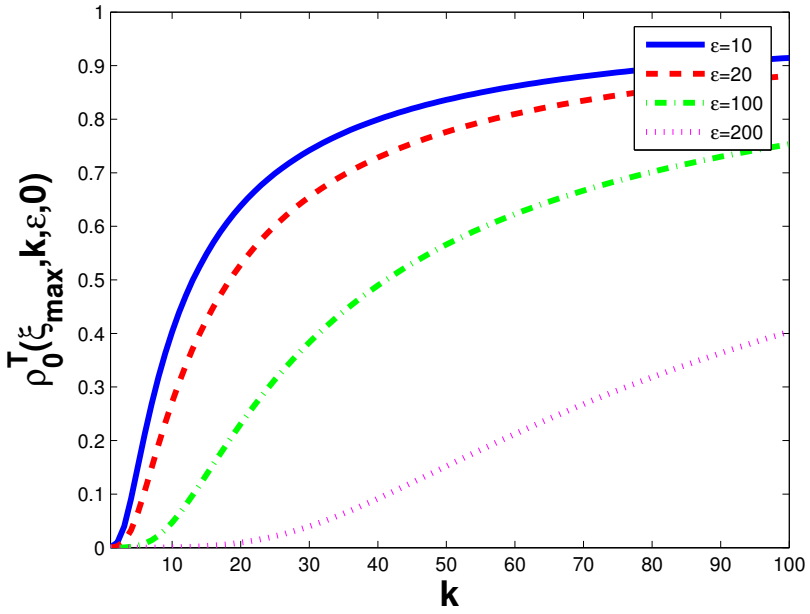

Figure 2-19: *Plot of the convergence rate* $\max_\xi |\rho_0^T(\xi, k, \epsilon, 0)|$ *(no overlap) as a function of* k*. Here we use* $h = \frac{\pi}{5k}$*.*

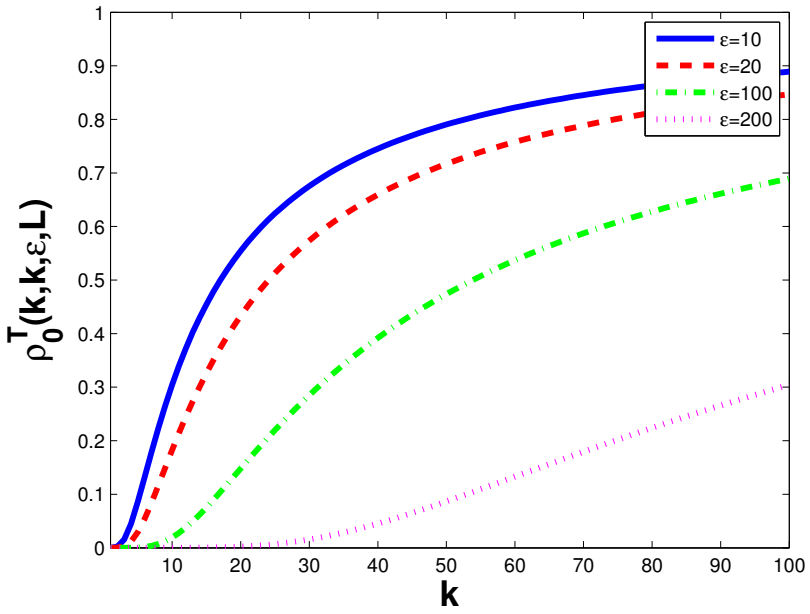

Figure 2-20: *Plot of the convergence rate* $\max_\xi |\rho_0^T(L)(\xi, k, \epsilon, L)|$ *as a function of* k*. Here we use* $h = \frac{\pi}{5k}$ *and an overlap of* $L = h$*.*

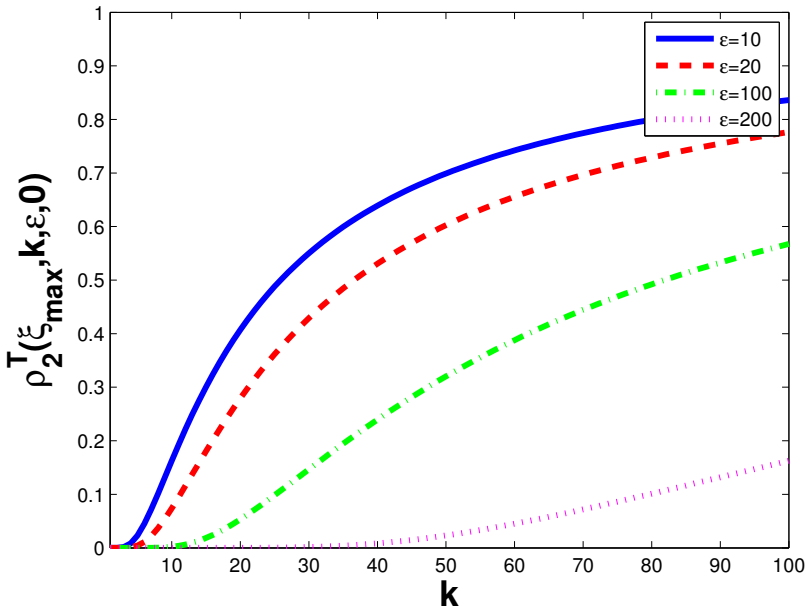

Figure 2-21: *Plot of the convergence rate* $\max_\xi |\rho_2^T(\xi, k, \epsilon, 0)|$ *(no overlap) as a function of* k. *Here we use* $h = \frac{\pi}{5k}$.

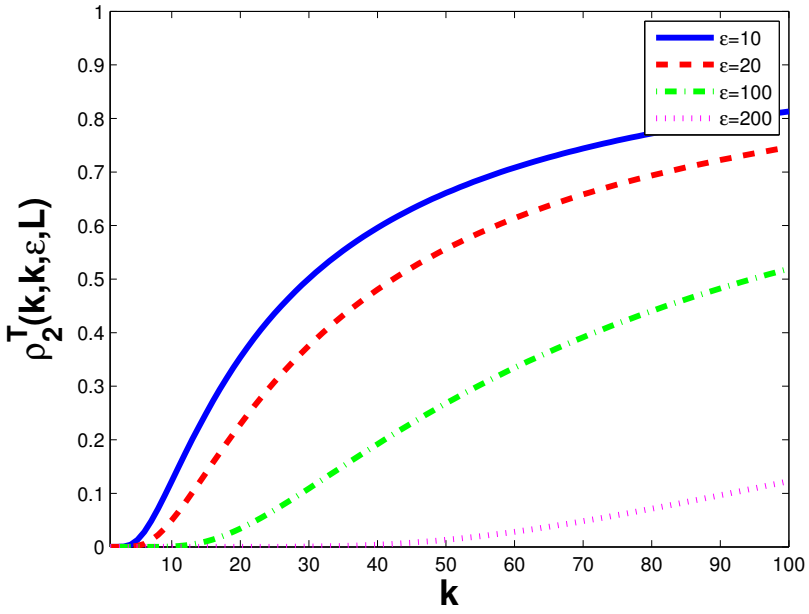

Figure 2-22: *Plot of the convergence rate*
$max_\xi |rho_2^T(L)(\xi, k, \epsilon, L)|$ *as a function of* k. *Here we use* $h = \frac{\pi}{5k}$ *and an overlap of* $L = h$.

CHAPTER 3

ANALYSIS FOR OPTIMISED TRANSMISSION CONDITIONS

In the previous chapter the multiplicative Schwarz method (2.17) for the Helmholtz equation (2.11) was introduced. It was shown that by choosing the multiplier in the interface conditions to be either the zeroth (2.56) or second order term (2.55) of the Taylor expansion of λ (2.28) in ξ we can achieve convergence of the Schwarz algorithm (2.17) but for which the convergence rate deteriorates as $k \to \infty$ This is not ideal and therefore we want to try and make the convergence rate depend less strongly on k, with independence of k being the overall goal. We now start by considering S of the form,

$$S_2^O = \alpha_1 + \alpha_2 \partial_{yy}^2, \ \alpha_1, \alpha_2 \in \mathbb{C}.$$

If we compare this to (2.55) we can see that all we have done is replace $\lambda(0, k, \epsilon)$ with a complex number which we will choose in a way which will improve convergence of (2.26). We make a simplification that $\alpha_1 = p(1+i)$ and $\alpha_2 = q(1+i)$ for $p, q \in \mathbb{R}_+$ which gives,

$$S_2^O = p(1+i) + q(1+i)\partial_{yy}^2. \tag{3.1}$$

We firstly look at the zeroth order term and how choosing p carefully can aid the convergence of (2.26). So we define our zeroth order term as

$$S_0^O = p(1+i). \tag{3.2}$$

If we now insert (3.2) into the convergence rate (2.22) this then gives,

$$\begin{aligned}
\left|\rho_0^O(\xi,k,\epsilon,p,0)\right| &= \left|\left(\frac{p+ip-\lambda(\xi,k,\epsilon)}{p+ip+\lambda(\xi,k,\epsilon)}\right)^2\right| \\
&= \left|\left(\frac{p+ip-(\lambda_R(\xi,k,\epsilon)+i\lambda_I(\xi,k,\epsilon))}{p+ip+(\lambda_R(\xi,k,\epsilon)+i\lambda_I(\xi,k,\epsilon))}\right)^2\right|, \text{ using (2.28)} \\
&= \left|\left(\frac{(p-\lambda_R(\xi,k,\epsilon))+i\,(p-\lambda_I(\xi,k,\epsilon))}{(p+\lambda_R(\xi,k,\epsilon))+i\,(p+\lambda_I(\xi,k,\epsilon))}\right)^2\right| \\
&= \frac{(p-\lambda_R(\xi,k,\epsilon))^2+(p-\lambda_I(\xi,k,\epsilon))^2}{(p+\lambda_R(\xi,k,\epsilon))^2+(p+\lambda_I(\xi,k,\epsilon))^2},
\end{aligned}$$

where we have taken the absolute value of ρ as in general it is complex. We therefore have that the convergence rate for the Schwarz algorithm (2.17) with $S = S_0^O$ is given by,

$$\begin{aligned}
\left|\rho_0^O(\xi,k,\epsilon,p,0)\right| &= \frac{(p-\lambda_R(\xi,k,\epsilon))^2+(p-\lambda_I(\xi,k,\epsilon))^2}{(p+\lambda_R(\xi,k,\epsilon))^2+(p+\lambda_I(\xi,k,\epsilon))^2}, \\
&= 1-4p\left(\frac{\lambda_R(\xi,k,\epsilon)+\lambda_I(\xi,k,\epsilon)}{p^2+2p\lambda_R(\xi,k,\epsilon)+\lambda_R(\xi,k,\epsilon)^2+p^2+2p\lambda_I(\xi,k,\epsilon)+\lambda_I(\xi)^2}\right), \\
&= 1-4F(\xi,k,\epsilon,p). \qquad (3.3)
\end{aligned}$$

where we now define the function,

$$F(\xi,k,\epsilon,p) = \frac{p(\lambda_R(\xi,k,\epsilon)+\lambda_I(\xi,k,\epsilon))}{(p+\lambda_R(\xi,k,\epsilon))^2+(p+\lambda_I(\xi,k,\epsilon))^2}, \qquad (3.4)$$

Therefore it is trivial to see that we require that $p \neq 0$, as if $p = 0$ then we see clearly that $\left|\rho_0^O(\xi,k,\epsilon,0,0)\right| = 1$. We also require that $p > 0$, as if $p < 0$ then it is clear from (3.3) that $\left|\rho_0^O(\xi,k,\epsilon,p,0)\right| > 1$. Therefore we choose $p \in \mathbb{R}_+$. We now prove that the right hand side of (3.3) is always less than one, and hence that the Schwarz algorithm (2.17) with the choice of (3.2) is convergent for all $\xi \in [\xi_{min}, \xi_{max}]$.

Theorem 3.1:
The Schwarz algorithm (2.17) *with transmission conditions of the form* (3.2) *is also convergent for all* $\xi \in [\xi_{min}, \xi_{max}] \subset [0,\infty)$*, and for all* $p \in \mathbb{R}_+$*. That is,*

$$\left|\rho_0^O(\xi,k,\epsilon,p,0)\right| < 1.$$

Proof. We recall the convergence rate (3.3),

$$\begin{aligned}
\left|\rho_0^O(\xi,k,\epsilon,p,0)\right| &= 1-4\left(\frac{p\lambda_R(\xi,k,\epsilon)+p\lambda_I(\xi,k,\epsilon)}{p^2+2p\lambda_R(\xi,k,\epsilon)+\lambda_R(\xi,k,\epsilon)^2+p^2+2p\lambda_I(\xi,k,\epsilon)+\lambda_I(\xi)^2}\right), \\
&= 1-4F\left(p,\xi,k,\epsilon\right).
\end{aligned}$$

Hence as $p > 0$ and as $\lambda_R, \lambda_I > 0$ it follows that,

$$\left|\rho_0^O(\xi, k, \epsilon, p, 0)\right| < 1.$$

Therefore the algorithm (2.17) with transmission conditions of the form (3.2) will always converge. □

The above result tells one that with this more general choice of condition (3.2) that we are guaranteed convergence of our algorithm as long as $p \in \mathbb{R}_+$. However, this does not tell one how fast we can expect to converge or how the number of iteration will be bounded with respect to k for instance. So how then do we choose p? It was shown in [21] that a good strategy to choose p is to solve the following optimisation problem. We find the maximum of the convergence rate (2.57) over all ξ in the relevant range $[\xi_{min}, \xi_{max}]$ and then use p to minimise this value. In doing so we find the choice of p which ensures near optimal convergence of the non overlapping Schwarz algorithm (2.17). Therefore we are now concerned with solving the following minimax problem,

$$\min_{p \in \mathbb{R}_+} \left(\max_{\xi_{min} \leq \xi \leq \xi_{max}} \left|\rho_0^O(\xi, k, \epsilon, p, 0)\right| \right), \tag{3.5}$$

where $\xi_{max} = \eta k$ is the largest frequency and $\xi_{min} \geq 0$ and,

$$\xi_{min} \leq k \leq \xi_{max}.$$

We return to the original minimax problem (3.5) and insert the convergence rate (3.3) into it. After expanding this expression we can see that it can we rewritten as a corresponding maximin problem,

$$\min_{p \in \mathbb{R}_+} \left(\max_{\xi_{min} \leq \xi \leq \xi_{max}} \left|\rho_0^O(\xi, k, \epsilon, p, 0)\right| \right) = 1 - 4 \left(\max_{p \in \mathbb{R}_+} \min_{\xi_{min} \leq \xi \leq \xi_{max}} F(\xi, k, \epsilon, p) \right). \tag{3.6}$$

Therefore we can solve the simpler maximin problem involving $F(\xi, k, \epsilon, p)$ to find the best choice of p.

3.1 Overview of results for zeroth order optimised transmission conditions without overlap

In this chapter we shall derive an optimised transmission condition for the Multiplicative Schwarz method (2.17) solving the Helmholtz problem with absorption (2.11). Before we state the main results, let us recall that the absolute value of the convergence rate of the Schwarz algorithm (3.3) can be written as,

$$\left|\rho_0^O\left(\xi,k,\epsilon,p,0\right)\right| = 1 - 4F\left(p,\xi,k,\epsilon\right). \tag{3.7}$$

where,

$$F\left(\xi,k,\epsilon,p\right) = \frac{p\left(\lambda_R\left(\xi,k,\epsilon\right)+\lambda_I\left(p,\xi,k,\epsilon\right)\right)}{\left(p+\lambda_R\left(\xi,k,\epsilon\right)\right)^2+\left(p+\lambda_I\left(\xi,k,\epsilon\right)\right)^2}, \tag{3.8}$$

and $\lambda_R(\xi,k,\epsilon) = \Re(\lambda(\xi,k,\epsilon))$, $\lambda_I(\xi,k,\epsilon) = \Im(\lambda(\xi,k,\epsilon))$, with

$$\lambda(\xi,k,\epsilon) = \sqrt{\xi^2-k^2+i\epsilon}.$$

Recall that we have chosen $\lambda(\xi,k,\epsilon)$ to lie in the first quadrant. The goal of this chapter is then to solve the following minimax problem,

$$\min_{p\in\mathbb{R}_+}\left(\max_{\xi_{min}\leq\xi\leq\xi_{max}}\rho_0^O(\xi,k,\epsilon,p,0)\right),$$

where we choose $\xi_{min}\geq 0$ and $\xi_{max} = 5k$ (see Remark 2.18 for a motivation). In solving (3.6) we find a p^* such that,

$$\max_{\xi_{min}\leq\xi\leq\xi_{max}}\rho_0^O(\xi,k,\epsilon,p^*,0) = \min_{p\in\mathbb{R}_+}\left(\max_{\xi_{min}\leq\xi\leq\xi_{max}}\rho_0^O(\xi,k,\epsilon,p,0)\right).$$

The advantage of computing such a p^* is that it allows us to consider the part of the convergence rate closest to 1 and minimise this, therefore improving the overall convergence rate. This p^* is then used as a parameter in the multiplicative Schwarz algorithm (2.17) to improve convergence. We note that given (3.7) we can rewrite (3.6) as the equivalent maximin problem,

$$\max_{p\in\mathbb{R}_+}\left(\min_{\xi_{min}\leq\xi\leq\xi_{max}}F\left(\xi,k,\epsilon,p\right)\right), \tag{3.9}$$

which simplifies the analysis somewhat. The main original results of this chapter are the following.

Theorem 3.2:
Let $\epsilon = k^{\delta}$, where $\delta \in [0, 2]$. Then for k large enough there exists a unique $p^* \in \mathbb{R}_+$ satisfying,

$$\min_{\xi_{min} \leq \xi \leq \xi_{max}} F(\xi, k, \epsilon, p^*) = \max_{p \in \mathbb{R}_+} \left(\min_{\xi_{min} \leq \xi \leq \xi_{max}} F\left(\xi, k, \epsilon, p\right) \right). \tag{3.10}$$

Moreover, as k increases, with $\xi_{max} = 5k$ then

$$p^* = 12^{\frac{1}{4}} k^{\frac{2+\delta}{4}} \left(1 + \mathcal{O}\left(k^{\frac{\delta-2}{2}}\right)\right) \tag{3.11}$$

The proof of this result can be found in Section 3.4. We can then use this result to show the following. The previous Theorem could also be proven for a general $\xi_{max} = \eta k$ (with $\eta > \sqrt{2}$) but we have chosen to prove it for this particular case to make the proof simpler.

Corollary 3.3:
Under the same assumptions as Theorem 3.2,

$$\max_{\xi_{min} \leq \xi \leq \xi_{max}} |\rho_0^O(\xi, k, \epsilon, p^*, 0)| = 1 - \frac{2}{3^{\frac{1}{4}}} k^{\frac{\delta-2}{4}} + \mathcal{O}\left(k^{\frac{\delta-2}{2}}\right). \tag{3.12}$$

The proof of this result can also be found in section 3.4. If we compare this result to that obtained using standard Taylor transmission conditions (2.58) we can clearly see that a more favourable convergence rate of $1 - \mathcal{O}\left(k^{\frac{\delta-2}{4}}\right)$ is obtained, in comparison to $1 - \mathcal{O}\left(k^{\delta-2}\right)$ for the Taylor method with no overlap (see Theorem 2.27). So as $k \to \infty$ the convergence rate of the Taylor method $\rho^T \to 1$ more quickly than that of the optimised method. Therefore the advantage of using this optimised condition (3.11) is that it leads to an algorithm with convergence rate which does not degrade as quickly when $k \to \infty$, as can be seen in the following Corollary.

Corollary 3.4:
Under the same assumptions as Theorem 3.2, the number of iterations of the Schwarz algorithm (2.17)*, with p^* chosen as in Theorem 3.2, needed to converge to a given tolerance τ is bounded below by,*

$$N_{iters} \geq \log\left(\frac{1}{\tau}\right) \frac{3^{\frac{1}{4}}}{4} k^{\frac{2-\delta}{4}} \tag{3.13}$$

This result is proved in section 3.4, and provides a lower bound on how the number of iterations of the Schwarz algorithm (2.17) should grow. Something worth noting here is that a choice of $\epsilon = k^2$, should lead to a number of iterations which does not grow with k. This is something which we shall be shown later in Chapter 4 with a numerical

implementation of the algorithm (2.17) and its use as a preconditioner for GMRES. In the following section we look at the behaviour of $F(\xi, k, \epsilon, p)$ with respect to ξ and p, this information will be needed later in the proof of the main results described previously.

3.2 Some elementary results about $F(\xi, k, \epsilon, p)$

3.2.1 Behaviour of $F(\xi, k, \epsilon, p)$ with respect to ξ

To solve the maximin problem (3.9) we must find the minimum of $F(\xi, k, \epsilon, p)$ over $\xi \in [\xi_{min}, \xi_{max}]$, and then the corresponding value of p which maximises it. Therefore we must compute the critical points of $F(\xi, k, \epsilon, p)$ with respect to both ξ and p, and study the behaviour of F with respect to both variables. Let us recall the definition of $F(\xi, k, \epsilon, p)$,

$$F(\xi, k, \epsilon, p) = \frac{p(\lambda_R(\xi, k, \epsilon) + \lambda_I(\xi, k, \epsilon))}{(p + \lambda_R(\xi, k, \epsilon))^2 + (p + \lambda_I(\xi, k, \epsilon))^2}.$$

In this subsection we shall study the variation of $F(\xi, k, \epsilon, p)$ with respect to ξ. Since in this discussion p, k, and ϵ are fixed, their dependence is dropped in the notation for brevity, i.e. we write $F(\xi) = F(\xi, k, \epsilon, p)$.

Lemma 3.5:
The first derivative of $F(\xi)$ with respect to ξ is given by,

$$\frac{\partial F}{\partial \xi}(\xi) = \frac{p\left(2p^2 - (\lambda_R(\xi) + \lambda_I(\xi))^2 - 2\lambda_R(\xi)\lambda_I(\xi)\right)(\lambda_R'(\xi) + \lambda_I'(\xi))}{\left((p + \lambda_R(\xi))^2 + (p + \lambda_I(\xi))^2\right)^2} \tag{3.14}$$

Proof. We simplify the notation further by suppressing the independent variable ξ. We differentiate (3.4) by using the quotient rule to obtain,

$$\begin{aligned}\frac{\partial F}{\partial \xi} &= \frac{p(\lambda_R' + \lambda_I')((p + \lambda_R)^2 + (p + \lambda_I)^2) - p(\lambda_R + \lambda_I)(2(p + \lambda_R)\lambda_R' + 2(p + \lambda_I)\lambda_I')}{\left((p + \lambda_R)^2 + (p + \lambda_I)^2\right)^2}, \\ &=: p\frac{T}{V}. \end{aligned} \tag{3.15}$$

The first step to simplify T is the following,

$$\begin{aligned} T &= (\lambda_R' + \lambda_I')(2p^2 + 2p(\lambda_R + \lambda_I) + \lambda_R^2 + \lambda_I^2) - 2(\lambda_R + \lambda_I)\left(p(\lambda_R' + \lambda_I') + \lambda_R\lambda_R' + \lambda_I\lambda_I'\right), \\ &= (\lambda_R' + \lambda_I')(2p^2 + \lambda_R^2 + \lambda_I^2) - 2(\lambda_R + \lambda_I)\left(\lambda_R\lambda_R' + \lambda_I\lambda_I'\right), \\ &= (\lambda_R' + \lambda_I')(2p^2 + \lambda_R^2 + \lambda_I^2 - 2\lambda_R\lambda_I) - 2\left(\lambda_R^2\lambda_R' + \lambda_I^2\lambda_I'\right). \end{aligned} \tag{3.16}$$

Moreover, we have that,

$$\begin{aligned}\lambda_R^2\lambda_R' + \lambda_I^2\lambda_I' &= \left(\lambda_R^2 + \lambda_I^2\right)\left(\lambda_R' + \lambda_I'\right) - \lambda_I^2\lambda_R' - \lambda_R^2\lambda_I', \\ &= \left(\lambda_R^2 + \lambda_I^2\right)\left(\lambda_I' + \lambda_R'\right) + \lambda_R\lambda_I\left(\lambda_I' + \lambda_R'\right), \end{aligned} \tag{3.17}$$

where the last step uses (2.39). If we substitute (3.17) into (3.16) we get,

$$\begin{aligned} T &= \left(\lambda_R' + \lambda_I'\right)\left(2p^2 - \lambda_R^2 - \lambda_I^2 - 4\lambda_R\lambda_I\right), \\ &= \left(\lambda_R' + \lambda_I'\right)\left(2p^2 - \left(\lambda_R + \lambda_I\right)^2 - 2\lambda_R\lambda_I\right). \end{aligned} \tag{3.18}$$

Which yields the result. □

Corollary 3.6:
$\frac{\partial F(\xi)}{\partial \xi} = 0$, *if and only if, either* $2p^2 = (\lambda_R(\xi)+\lambda_I(\xi))^2+2\lambda_R(\xi)\lambda_I(\xi)$ *or* $\lambda_R'(\xi) = -\lambda_I'(\xi)$.

Proof. The result follows immediately from (3.14). □

We can then prove the following result concerning the critical points of $F(\xi, k, \epsilon, p)$ as a function of ξ.

Theorem 3.7:
The critical points of $F(\xi, k, \epsilon, p)$ *behave according to the following two statements:*

(i) For all $p \in \left(\sqrt{\frac{3\epsilon}{2}}, \infty\right)$, $F(\xi, k, \epsilon, p)$ *has exactly one local minimum with respect to* ξ, *which occurs at* $\xi = k$ *and all* $\xi \in (\xi_{min}, \xi_{max})$ *and all* $\epsilon > 0$.
(ii) For all $p \in \left(0, \sqrt{\frac{3\epsilon}{2}}\right)$, $F(\xi, k, \epsilon, p)$ *has exactly one local maximum with respect to* ξ, *which occurs at* $\xi = k$ *and all* $\xi \in (\xi_{min}, \xi_{max})$ *and all* $\epsilon > 0$.

Proof. From Corollary 3.6 we know that the zeros of $\frac{\partial F}{\partial \xi}$ occur when

$$\begin{aligned} &\text{Case 1: } \lambda_R'(\xi) = -\lambda_I'(\xi), \text{ or,} \\ &\text{Case 2: } 2p^2 = (\lambda_R(\xi) + \lambda_I(\xi))^2 + 2\lambda_R(\xi)\lambda_I(\xi). \end{aligned}$$

As we wish to show whether these are local maxima or minima we first compute the second derivative of F with respect to ξ. If we recall (3.15) that $F' = p\frac{T}{V}$, where T, V are given in (3.15),(3.18), and then applying the quotient rule again on this we obtain,

$$F'' = p\left(\frac{T'V - TV'}{V^2}\right).$$

But we already know that if we are at a zero of $\frac{\partial F}{\partial \xi}$, then $T = 0$, and also that $V > 0$ and $p > 0$. Therefore the sign of F'' at such a zero is governed by that of T'. Computing

T' we obtain,

$$T' = (-2(\lambda_R\lambda_R' + \lambda_I\lambda_I') - 4(\lambda_R\lambda_I' + \lambda_I\lambda_R')(\lambda_R' + \lambda_I') + \left(2p^2 - (\lambda_R + \lambda_I)^2 - 2\lambda_R\lambda_I\right)(\lambda_R'' + \lambda_I''),$$

Then as $\lambda_R\lambda_I' + \lambda_I\lambda_R' = 0$ (by (2.39)) we obtain

$$T' = -2(\lambda_R\lambda_R' + \lambda_I\lambda_I')(\lambda_R' + \lambda_I') + \left(2p^2 - (\lambda_R + \lambda_I)^2 - 2\lambda_R\lambda_I\right)(\lambda_R'' + \lambda_I''). \quad (3.19)$$

Referring back to Case 1 and Case 2 above we examine the sign of T' in each of these cases.

Case 1: $\lambda_R'(\xi) = -\lambda_I'(\xi)$
In this case (3.19) immediately simplifies to

$$T' = \left(2p^2 - (\lambda_R + \lambda_I)^2 - 2\lambda_R\lambda_I\right)(\lambda_R'' + \lambda_I'').$$

We can make some further simplifications. Since, by Lemma 2.17, $\xi = k$ we have by Remark 2.12 that $\lambda_R = \lambda_I = \sqrt{\frac{\epsilon}{2}}$, and therefore,

$$T' = \left(2p^2 - 3\epsilon\right)(\lambda_R'' + \lambda_I''). \quad (3.20)$$

Consider now the sign of $(\lambda_R'' + \lambda_I'')$. Differentiating (2.39) with respect to ξ, we obtain

$$(\lambda_R\lambda_I' + \lambda_I\lambda_R')' = 0,$$
$$\text{hence, } \lambda_R'\lambda_I' + \lambda_R\lambda_I'' + \lambda_I'\lambda_R' + \lambda_I\lambda_R'' = 0.$$
$$\text{Thus, } 2\lambda_R'\lambda_I' + (\lambda_R\lambda_I'' + \lambda_I\lambda_R'') = 0, \text{ and since } \lambda_R = \lambda_I = \sqrt{\frac{\epsilon}{2}},$$
$$\text{this then implies that } 2\lambda_R'\lambda_I' + \sqrt{\frac{\epsilon}{2}}(\lambda_I'' + \lambda_R'') = 0.$$

Rearranging this we get that,

$$(\lambda_R'' + \lambda_I'') = -2\sqrt{\frac{2}{\epsilon}}\lambda_R'\lambda_I'.$$

By Lemma 2.16 we have $\operatorname{sgn}(\lambda_R'\lambda_I') = -1$ ($\lambda_R', \lambda_I' \neq 0$, and it therefore follows that as $\epsilon > 0$ then

$$\operatorname{sgn}\left(-2\sqrt{\frac{2}{\epsilon}}\lambda_R'\lambda_I'\right) = \operatorname{sgn}\left(-2\sqrt{\frac{2}{\epsilon}}\right)\operatorname{sgn}\left(\lambda_R', \lambda_I'\right)$$
$$= 1.$$

It follows then that

$$(\lambda_R'' + \lambda_I'') > 0. \tag{3.21}$$

Hence,

$$T' = \underbrace{\left(2p^2 - 3\epsilon\right)}_{>0 \text{ if } p > \sqrt{3\epsilon/2}} \underbrace{(\lambda_R'' + \lambda_I'')}_{>0} > 0.$$

To summarise Case 1, the only solution is $\xi = k$ which is a local minimum if $p \in (\sqrt{\frac{3\epsilon}{2}}, \infty)$. On the other hand $\xi = k$ is a local maximum of $F(\xi, k, \epsilon, p)$ if $p < \sqrt{\frac{3\epsilon}{2}}$.

Case 2: $2p^2 = (\lambda_R(\xi) + \lambda_I(\xi))^2 + 2\lambda_R(\xi)\lambda_I(\xi)$
We now consider the other case when $2p^2 = (\lambda_R + \lambda_I)^2 + 2\lambda_R\lambda_I$ and prove that there are either no solutions to this equation (when $p < \sqrt{\frac{3\epsilon}{2}}$) or there are two solutions which lie on either side of $\xi = k$, and hence these are local maxima of T (when $p > \sqrt{\frac{3\epsilon}{2}}$).
Let us first define,

$$M(\xi) = (\lambda_R(\xi) + \lambda_I(\xi))^2 + 2\lambda_R(\xi)\lambda_I(\xi). \tag{3.22}$$

Differentiating, we obtain

$$\begin{aligned} M' &= 2(\lambda_R + \lambda_I)(\lambda_R' + \lambda_I') + \underbrace{2(\lambda_R'\lambda_I + \lambda_I'\lambda_R)}_{=0, \text{ by (2.39)}}, \\ &= 2(\lambda_R + \lambda_I)(\lambda_R' + \lambda_I'). \end{aligned}$$

Therefore, by Proposition 2.13 and Lemma 2.17, the only zero of $M'(\xi)$ is $\xi = k$. On examining the second derivative of M, using Proposition 2.13 and (3.21), we have at $\xi = k$

$$M''(k) = 2\underbrace{(\lambda_R'(k) + \lambda_I'(k))^2}_{>0} + 2\underbrace{(\lambda_R(k) + \lambda_I(k))(\lambda_R''(k) + \lambda_I''(k))}_{>0} > 0.$$

Thus there exists a single local minimum of $M(\xi)$ at $\xi = k$. If we recall that $\lambda_R = \lambda_I = \sqrt{\epsilon/2}$ at $\xi = k$ then we have $M(k) = 3\epsilon$. We now consider the behaviour of $M(\xi)$ as $\xi \to \infty$ If we recall (2.28) and (2.29) then it follows that,

$$\begin{aligned} \xi^2 - k^2 + i\epsilon &= (\lambda_R(\xi) + i\lambda_I(\xi))^2, \\ &= \lambda_R^2(\xi) - \lambda_I^2(\xi) + 2i\lambda_R(\xi)\lambda_I(\xi). \end{aligned}$$

Then taking the real part we obtain

$$\lambda_R^2(\xi) - \lambda_I^2(\xi) = \xi^2 - k^2.$$

If we recall Proposition 2.13 then the following is true,

$$\begin{aligned} \lambda_R^2(\xi) &> \lambda_R^2(\xi) - \lambda_I^2(\xi), \text{ and it follows that,} \\ &= \xi^2 - k^2 \to \infty, \quad \text{as } \xi \to \infty. \end{aligned}$$

Therefore as $\lambda_R(\xi) \to \infty$ as $\xi \to \infty$ and if we recall (3.22) and Proposition 2.13, it follows that $M(\xi) \to \infty$ as $\xi \to \infty$, and $M(\xi)$ behaves as shown numerically in Figure 3-1. Now recall (see statement of Case 2 above), that we interested in solutions to $2p^2 = M(\xi)$ for fixed p. It is clear that if $2p^2 < 3\epsilon$ then there is no solution. If $2p^2 > 3\epsilon$, then there are at most two solutions one on each side of $\xi = k$. Therefore as argued above, they must be local maxima of F. □

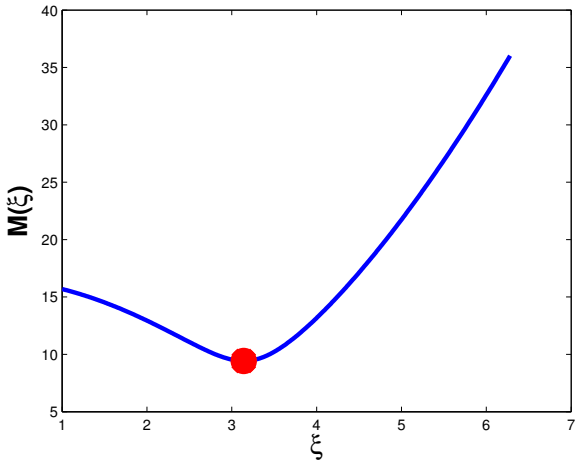

Figure 3-1: *Plot of $M(\xi)$ for $k = \epsilon = \pi$ and $\xi \in [1, 2k]$. The circle indicates $M = 3\epsilon$ when $\xi = k$.*

In Figures 3-2, and 3-3 we plot $F(\xi, k, \epsilon, p)$ against ξ for ξ in a fixed range for fixed k and ϵ with p chosen to show each of the cases in Theorem 3.6. In Figure 3-2 $p = k > \sqrt{\frac{3\epsilon}{2}}$ so $F(\xi, k, \epsilon, p)$ should achieve a minimum at $\xi = k$ according to the Theorem, and this is what is shown numerically. Similarly in Figure 3-3 we choose $p = \sqrt{\frac{3\epsilon}{4}} < \sqrt{\frac{3\epsilon}{2}}$ and observe that a maximum occurs at $\xi = k$ and no other critical points, as proved theoretically.

Therefore we can conclude from Theorem 3.7 that the function $F(\xi, k, \epsilon, p)$, as a function of ξ, has only one local minimum at $\xi = k$ where $p > \sqrt{\frac{3\epsilon}{2}}$ and no local minima when $p < \frac{3\epsilon}{2}$. From this we have the following Theorem which gives details of the

possible location of the minimum of F in ξ for fixed p, k and ϵ.

Theorem 3.8:
For each $p \in \mathbb{R}_+$,

$$\min_{\xi_{min}, \leq \xi \leq \xi_{max}} F(\xi, k, \epsilon, p) = \min\{F(\xi_{min}, k, \epsilon, p), F(k, k, \epsilon, p), F(\xi_{max}, k, \epsilon, p)\} \quad (3.23)$$

Proof. By Theorem 3.7 it is clear that the only possible local minimum of $F(\xi, k, \epsilon, p)$ with respect to ξ is k. Therefore to compute the global minimum we need only to look at $\xi = k$ and the end points $\xi = \xi_{min}$ and $\xi = \xi_{max}$. □

3.2.2 Behaviour of $F(\xi, k, \epsilon, p)$ with respect to p

Up until now we have examined $F(\xi, k, \epsilon, p)$ as a function of ξ, and its critical points with respect to ξ. Since p is the solution of (3.9) it is necessary for the proof of Theorem 3.2 for us now to examine the behaviour of $F(\xi, k, \epsilon, p)$ with respect to p for fixed ξ, k and ϵ. Recalling the definition of $F(\xi, k, \epsilon, p)$, stated in (3.8) and recalling Proposition 2.13, it is trivial to see that for fixed ξ, k, ϵ $F(\xi, k, \epsilon, p) \to 0$ when $p \to 0$ or $p \to \infty$. We now examine the possible critical points of $F(\xi, k, \epsilon, p)$ with respect to p. Here ξ, k, and ϵ are fixed so their dependence is dropped in the notation for brevity.

Lemma 3.9:
For fixed ξ, k, and ϵ, then $F(\xi, k, \epsilon, p)$ has a single critical point,

$$p^c = \sqrt{\frac{\lambda_R^2(\xi, k, \epsilon) + \lambda_I^2(\xi, k, \epsilon)}{2}}, \quad (3.24)$$

which is the global maximum of $F(\xi, k, \epsilon, p)$, as a function of p.

Proof. Differentiating F with respect to p using the quotient rule and simplifying we have

$$\frac{\partial F}{\partial p} = \frac{\left(-2p^2 + \lambda_R^2 + \lambda_I^2\right)(\lambda_R + \lambda_I)}{\left((p + \lambda_R)^2 + (p + \lambda_I)^2\right)^2} \quad (3.25)$$

Hence, recalling Proposition 2.13, $\frac{\partial F}{\partial p} = 0$ if and only if $2p^2 = \lambda_R^2 + \lambda_I^2$, and thus there is a single critical point given by (3.24), and $F(\xi, k, \epsilon, p^*) > 0$. Also recalling that as $F(\xi, k, \epsilon, p) \to 0$ when either $p \to 0$ or $p \to +\infty$, it follows that the critical point p^c must be a global maximum of $F(\xi, k, \epsilon, p)$. □

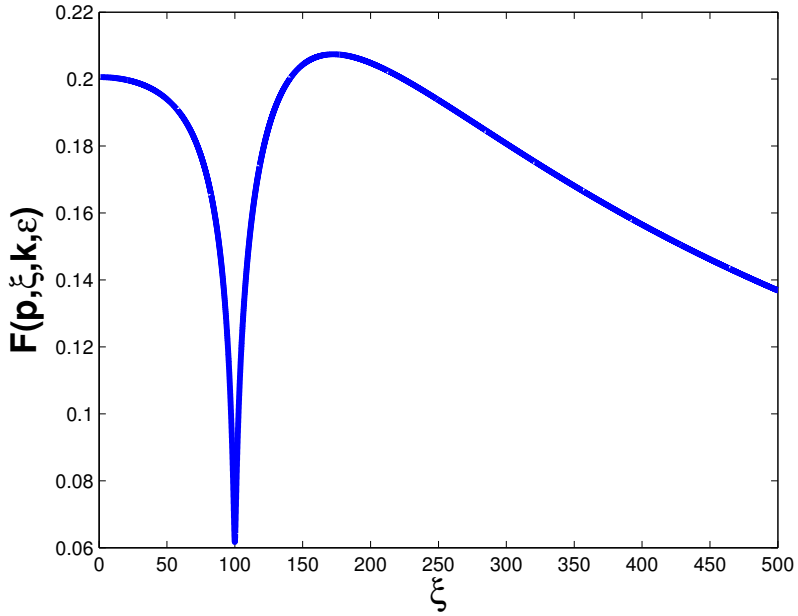

Figure 3-2: *Plot of the function* $F(\xi, k, \epsilon, p)$ *for* $k = 100$, $\epsilon = k$, $\xi = [0, 5k]$, *and a choice of* $p = k$.

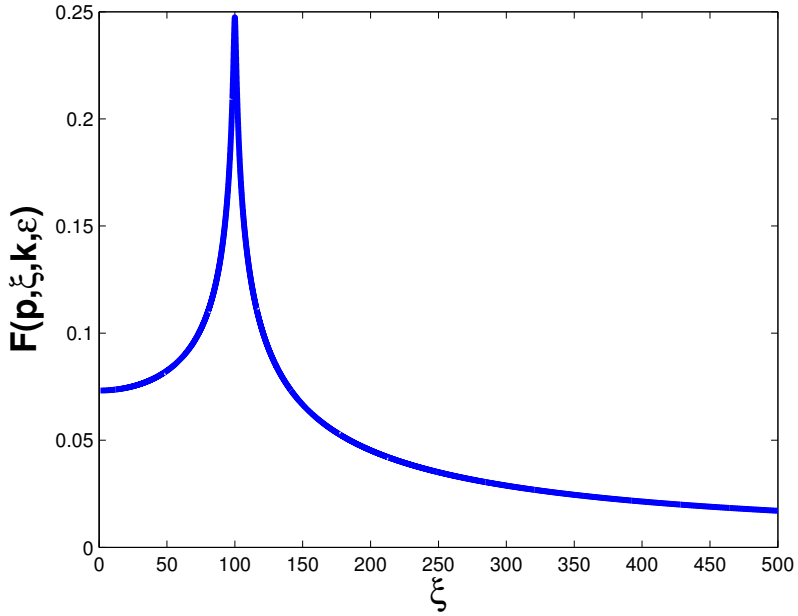

Figure 3-3: *Plot of the function* $F(\xi, k, \epsilon, p)$ *for* $k = 100$, $\epsilon = k$, $\xi = [0, 5k]$, *and a choice of* $p = \sqrt{\frac{3\epsilon}{4}}$.

The main result of the previous subsection was Theorem 3.8 which told us that the minimum of $F(\xi, k, \epsilon, p)$ with respect to ξ occurs at either $\xi = \xi_{min}$, k or ξ_{max}. Therefore to solve our maximin problem (3.9) we need only concern ourselves with $F(\xi, k, \epsilon, p)$ at these three values of ξ. We now examine the behaviour of p^c at these three values of ξ for k sufficiently large. This is necessary for the proof of Theorem 3.2 as we shall see later. We remind the reader (see Remark 2.18) that we fix

$$\xi_{min} \geq 0, \quad \text{fixed independent of } k \tag{3.26a}$$

$$\xi_{max} = \eta k, \quad \text{for } \eta > \sqrt{2} \tag{3.26b}$$

$$\text{and } \gamma = \sqrt{\eta^2 - 1} > 1. \tag{3.26c}$$

Lemma 3.10:
For $\epsilon = k^\delta$ (where $\delta \in [0, 2]$) the critical point of $F(\xi, k, \epsilon, p)$ as a function of p is given by the following expressions at the corresponding fixed ξ as $k \to \infty$

$$p^c_{\xi_{min}} = \frac{k}{\sqrt{2}} + \mathcal{O}\left(k^{2(\delta-2)}\right), \ \textit{when } \xi = \xi_{min}. \tag{3.27a}$$

$$p^c_k = \frac{k^{\frac{\delta}{2}}}{\sqrt{2}}, \ \textit{when } \xi = k. \tag{3.27b}$$

$$p^c_{\xi_{max}} = \frac{\gamma k}{\sqrt{2}} + \mathcal{O}\left(k^{2(\delta-2)}\right), \ \textit{when } \xi = \xi_{max}. \tag{3.27c}$$

Proof. If we recall the definition of p_c given by equation (3.24) then we simply have to substitute the specified value of ξ into $\lambda_R(\xi, k, \epsilon)$ and $\lambda_I(\xi, k, \epsilon)$. For $\xi = k$ we have (see Remark 2.12)

$$\lambda_R(k, k, \epsilon) = \lambda_I(k, k\epsilon) = \sqrt{\frac{\epsilon}{2}} = \sqrt{\frac{k^\delta}{2}},$$

and hence by (3.24),

$$p^c_k = \frac{k^{\frac{\delta}{2}}}{\sqrt{2}}.$$

However, for $\xi = \xi_{min}$ and $\xi = \xi_{max}$ we use the asymptotic expressions for $\lambda_R(\xi, k, \epsilon)$ and $\lambda_I(\xi, k, \epsilon)$ given in Lemmas 2.19 and 2.20 for k increasing. Substituting these results into (3.24) gives (3.27a) and (3.27c). □

3.3 The solution of the maximin problem

Throughout this section it is assumed that $\epsilon = k^{\delta}$, where $\delta \in [0, 2]$ and that the maximum and minimum values of ξ as given in (3.26). Some investigations are theoretical and some are numerical. In all numerical examples we have taken $\eta = 5$ so $\gamma = \sqrt{24}$. Recalling Theorem 3.8 we start by calculating the points in p where $F(\xi_{min}, k, \epsilon, p)$, $F(k, k, \epsilon, p)$ and $F(\xi_{max}, k, \epsilon, p)$ intersect, and then derive an inequality relating their magnitudes and those of the critical points (3.27a), (3.27b) and (3.27c).

3.3.1 Strategy for computing the solution of the maximin problem

We start with an example illustrating the strategy to solve the maximin problem (3.9). If we recall Theorem 3.8 then our maximin problem can be simplified to the following,

$$\max_{p \in \mathbb{R}_+} \min \Big(F(\xi_{min}, k, \epsilon, p), F(k, k, \epsilon, p), F(\xi_{max}, k, \epsilon, p) \Big). \tag{3.28}$$

We now consider how the three functions in (3.28) behave with respect to p for fixed k and ϵ. Recalling Lemma 3.9 we know that $F(\xi, k, \epsilon, p)$ has a single maximum with respect to p. Therefore let us consider a particular example of the three functions $F(\xi, k, \epsilon, p)$ in (3.28) and plot them to see where the solution of (3.28) occurs. In Figure 3-4 we plot these functions for $k = 100$, $\epsilon = k$, $\xi_{min} = \pi$ and $\xi_{max} = 5k$. We observe that each function intersects each of the others exactly once. It is actually trivial to show that there is only one $p > 0$ which satisfies

$$F(\xi_1, k, \epsilon, p) = F(\xi_2, k, \epsilon, p), \tag{3.29}$$

where $\xi_1, \xi_2 \in \{\xi_{min}, k, \xi_{max}\}$ and $\xi_1 \neq \xi_2$. One shows this by observing that (3.29) is a quadratic equation for p for which there is only one real solution. We denote the three points of intersection as the following,

$$p_{k,\xi_{min}} = \text{ the unique } p \in \mathbb{R}_+ \text{ for which } F(k, k, \epsilon, p) = F(\xi_{min}, k, \epsilon, p), \tag{3.30a}$$

$$p_{k,\xi_{max}} = \text{ the unique } p \in \mathbb{R}_+ \text{ for which } F(k, k, \epsilon, p) = F(\xi_{max}, k, \epsilon, p), \tag{3.30b}$$

$$p_{\xi_{min},\xi_{max}} = \text{ the unique } p \in \mathbb{R}_+ \text{ for which } F(\xi_{min}, k, \epsilon, p) = F(\xi_{max}, k, \epsilon, p), \tag{3.30c}$$

One can observe from Figure 3-4 that the solution of (3.28) is in fact the point $p_{k,\xi_{max}}$ which is where the functions $F(k, k, \epsilon, p)$ and $F(\xi_{max}, k, \epsilon, p)$ (the dashed line and the solid line respectively) intersect. One can reason that this is the solution of (3.28) as

when $p < p_{k,\xi_{max}}$ then one can see that,

$$\min\left(F\left(\xi_{min},k,\epsilon,p\right),F\left(k,k,\epsilon,p\right),F\left(\xi_{max},k,\epsilon,p\right)\right)=F\left(\xi_{max},k,\epsilon,p\right),$$

where $F\left(\xi_{max},k,\epsilon,p\right)$ is increasing when $p < p_{k,\xi_{max}}$. We also observe that when $p > p_{k,\xi_{max}}$,

$$\min\left(F\left(\xi_{min},k,\epsilon,p\right),F\left(k,k,\epsilon,p\right),F\left(\xi_{max},k,\epsilon,p\right)\right)=F\left(k,k,\epsilon,p\right),$$

where $F\left(k,k,\epsilon,p\right)$ is decreasing when $p > p_{k,\xi_{max}}$. This however only proves that the solution of (3.28) is $p_{k,\xi_{max}}$ for this particular choice of k and ϵ in Figure 3-4.

In the following subsections we shall prove the necessary results to solve (3.28) for the asymptotic range of ϵ considered and k increasing. The required results are; the points of intersection in p, and the corresponding value of $F\left(\xi,k,\epsilon,p\right)$ at these intersections. These results then allow us to construct an argument similar to that used in our example for Figure 3-4.

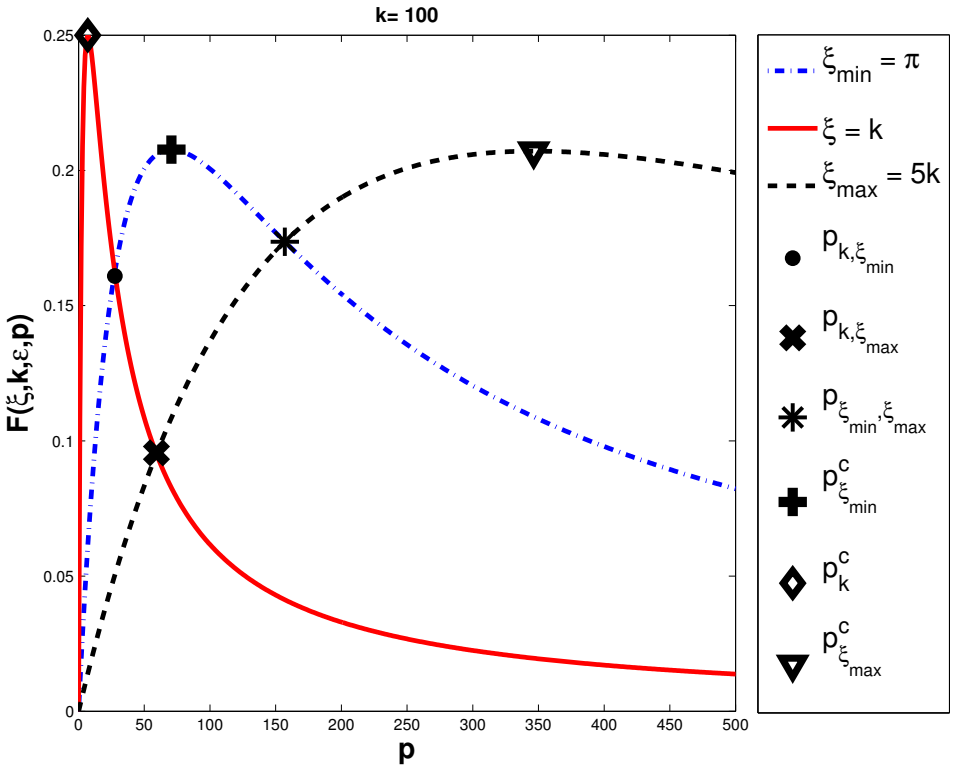

Figure 3-4: *Plot of the functions* $F(\pi,k,\epsilon,p)$, $F\left(k,k,\epsilon,p\right)$ *and* $F(5k,k,\epsilon,p)$ *for* $k = 100$, $\epsilon = k$, $p = [0,5k]$.

3.3.2 Analysis of the relative positions of $p_{k,\xi_{min}}$, $p_{k,\xi_{max}}$, $p_{\xi_{min},\xi_{max}}$, p_k^c, $p_{\xi_{min}}^c$ and $p_{\xi_{max}}^c$

Recalling the definitions of the intersection (3.30) and the critical points (3.27) in this subsection we examine the relative positions of each of these values of p for k sufficiently large. This is achieved in Theorem 3.14 where we establish an inequality relating the relative positions of these quantities for increasing k. After this we calculate the corresponding values of $F(\xi, k, \epsilon, p)$ at each of $p_{k,\xi_{min}}$, $p_{k,\xi_{max}}$ and $p_{\xi_{min},\xi_{max}}$ and provide an inequality between these values of $F(\xi, k, \epsilon, p)$, given in Theorem 3.18. These two inequalities are then used together with results from section 3.2.2 to obtain the solution of problem(3.9). The main results, Theorem 3.2 and Corollary 3.3 are proved in Section 3.4.

We start by computing $p_{k,\xi_{max}}$ the solution of $F(k, k, \epsilon, p) = F(\xi_{max}, k, \epsilon, p)$ for increasing k

Lemma 3.11:
The value of $p \in \mathbb{R}_+$ which solves $F(k, k, \epsilon, p) = F(\xi_{max}, k, \epsilon, p)$, where $\epsilon = k^\delta$, for k sufficiently large enough is given by,

$$p_{k,\xi_{max}} = \left(\frac{\gamma}{\sqrt{2}}\right)^{\frac{1}{2}} k^{\frac{2+\delta}{4}} \left(1 + \mathcal{O}\left(k^{\frac{\delta-2}{2}}\right)\right). \tag{3.31}$$

Proof. One can show that the only possible positive and real solution of $F(k, k, \epsilon, p) = F(\xi_{max}, k, \epsilon, p)$ is given by the formula,

$$p_{k,\xi_{max}} = \sqrt{\frac{\left(\lambda_R^2(\xi_{max}) + \lambda_I^2(\xi_{max})\right) - \frac{\lambda_R(\xi_{max})+\lambda_I(\xi_{max})}{\lambda_R(k)+\lambda_I(k)}\left(\lambda_R^2(k) + \lambda_I^2(k)\right)}{2\left(\frac{\lambda_R(\xi_{max})+\lambda_I(\xi_{max})}{\lambda_R(k)+\lambda_I(k)} - 1\right)}}, \tag{3.32}$$

where for this equation we have dropped the k and ϵ dependence to make the equation shorter. One can then recall the leading order terms of (2.43) and (2.37), and inserting then into (3.32) to obtain,

$$p_{k,\xi_{max}} = \sqrt{\gamma k^2 \frac{\left(\gamma + \frac{k^{2(\delta-2)}}{4\gamma^3} - \frac{k^{\frac{\delta-2}{2}}}{\sqrt{2}} - \frac{k^{\frac{3(\delta-2)}{2}}}{2\sqrt{2}\gamma^2}\right)}{2\left(\frac{\gamma k + \frac{k^{\delta-1}}{2\gamma}}{\sqrt{2}k^{\frac{\delta}{2}}} - 1\right)}} + \text{ Higher order terms.}$$

We proceed by multiplying the numerator and denominator by $\sqrt{2}k^{\frac{\delta}{2}}$ and dividing both

by k to give,

$$p_{k,\xi_{max}} = \sqrt{\frac{\gamma}{\sqrt{2}} k^{\frac{2+\delta}{2}} \left(\frac{\gamma + \frac{k^{2(\delta-2)}}{4\gamma^3} - \frac{k^{\frac{\delta-2}{2}}}{\sqrt{2}} - \frac{k^{\frac{3(\delta-2)}{2}}}{2\sqrt{2}\gamma^2}}{\gamma + \frac{k^{\delta-2}}{2\gamma} - \sqrt{2}k^{\frac{\delta-2}{2}}} \right)}.$$

Taking out a factor of $\left(\frac{\gamma}{\sqrt{2}}\right)^{\frac{1}{2}} k^{\frac{2+\delta}{4}}$ from the square root in the above equation this gives,

$$p_{k,\xi_{max}} = \left(\frac{\gamma}{\sqrt{2}}\right)^{\frac{1}{2}} k^{\frac{2+\delta}{4}} \sqrt{\frac{\gamma + \frac{k^{2(\delta-2)}}{4\gamma^3} - \frac{k^{\frac{\delta-2}{2}}}{\sqrt{2}} - \frac{k^{\frac{3(\delta-2)}{2}}}{2\sqrt{2}\gamma^2}}{\gamma + \frac{k^{\delta-2}}{2\gamma} - \sqrt{2}k^{\frac{\delta-2}{2}}}}. \tag{3.33}$$

Then as $p_{k,\xi_{max}}$ must be greater than zero we need to check that the square root in (3.33) is strictly greater than zero. We define the square root in (3.33) as the following

$$f_{p_{k,\xi_{max}}}(k,\delta,\gamma) = \sqrt{\frac{\gamma + \frac{k^{2(\delta-2)}}{4\gamma^3} - \frac{k^{\frac{\delta-2}{2}}}{\sqrt{2}} - \frac{k^{\frac{3(\delta-2)}{2}}}{2\sqrt{2}\gamma^2}}{\gamma + \frac{k^{\delta-2}}{2\gamma} - \sqrt{2}k^{\frac{\delta-2}{2}}}}. \tag{3.34}$$

If $\delta \in [0,2)$ then one can show that by taking the Taylor expansion of $f_{p_{k,\xi_{max}}}(k,\delta,\gamma)$ for $k \to \infty$ that,

$$f_{p_{k,\xi_{max}}}(k,\delta,\gamma) = 1 + \mathcal{O}\left(k^{\frac{1}{2}(\delta-2)}\right).$$

Hence the result (3.31). However if $\delta = 2$ then,

$$f_{p_{k,\xi_{max}}}(k,2,\gamma) = 1.079431\ldots, \tag{3.35}$$

when $\gamma = \sqrt{24}$. Therefore (3.33) is real and greater than zero for all $\delta \in [0,2]$. □

The proofs for Lemmas 3.12 and 3.13 are similar, therefore we present the results.

Lemma 3.12:
The value of $p \in \mathbb{R}_+$ which solves $F(k,k,\epsilon,p) = F(\xi_{min},k,\epsilon,p)$, where $\epsilon = k^\delta$, for k sufficiently large enough is

$$p_{k,\xi_{min}} = \frac{k^{\frac{2+\delta}{4}}}{2^{\frac{1}{4}}}\left(1 + \mathcal{O}\left(k^{\frac{\delta-2}{2}}\right)\right). \tag{3.36}$$

Lemma 3.13:
The value of $p \in \mathbb{R}_+$ which solves $F(k, \xi_{max}, \epsilon, p) = F(\xi_{min}, k, \epsilon, p)$, where $\epsilon = k^\delta$, for k sufficiently large enough is

$$p_{\xi_{min},\xi_{max}} = \sqrt{\frac{\gamma}{2}} k \left(1 + \mathcal{O}\left(k^{\delta-2}\right)\right). \tag{3.37}$$

We now form an inequality from the results of Lemmas 3.11, 3.12 and 3.13 and the critical points (3.27a), (3.27b), (3.27c) for k sufficiently large.

Theorem 3.14:
For $\epsilon = k^\delta$, where $\delta \in [0, 2)$, and k sufficiently large enough,

$$p_k^c < p_{k,\xi_{min}} < p_{k,\xi_{max}} < p_{\xi_{min}}^c < p_{\xi_{min},\xi_{max}} < p_{\xi_{max}}^c. \tag{3.38}$$

For $\delta = 2$ and k sufficiently large we find that,

$$p_k^c < p_{\xi_{min}}^c < p_{k,\xi_{min}} < p_{k,\xi_{max}} < p_{\xi_{min},\xi_{max}} < p_{\xi_{max}}^c. \tag{3.39}$$

Proof. We consider first the case when $\delta \in [0, 2)$. Firstly by simply recalling (3.31), (3.36) and (3.37) it is clear that both $p_{k,\xi_{max}}$ and $p_{k,\xi_{min}}$ behave like $k^{\frac{1}{4}(2+\delta)}$, and $p_{\xi_{min},\xi_{max}}$ behaves asymptotically like k. If we recall (3.27a), (3.27b), (3.27c) it is clear that both $p_{\xi_{min}}^c$ and $p_{\xi_{max}}^c$ grow like k, and p_k^c grows like $k^{\frac{\delta}{2}}$. If we collect this information we observe that we have one term which is of the order $k^{\frac{\delta}{2}}$, two which are of order $k^{\frac{1}{4}(2+\delta)}$, and three which are of order k. Then since $\delta = [0, 2)$ it is clear that $k^{\frac{\delta}{2}} < k^{\frac{1}{4}(2+\delta)} < k$, and therefore we know where to begin in constructing an inequality. We can see that as $p_k^c \sim k^{\frac{\delta}{2}}$ it is clearly the smallest of all six for all $\delta \in [0, 2)$. The terms which are next largest will be $p_{k,\xi_{max}}$ and $p_{k,\xi_{min}}$ which behave like $k^{\frac{1}{4}(2+\delta)}$. If recall (3.31) and (3.36) then it is clear that they only differ by a factor of $\sqrt{\gamma}$ that is $p_{k,\xi_{max}} = \sqrt{\gamma} p_{k,\xi_{min}}$. Then as $\sqrt{\gamma} > 1$ it is clear that $p_{k,\xi_{max}} > \sqrt{\gamma} p_{k,\xi_{min}}$. Finally we must consider $p_{\xi_{min},\xi_{max}}$, $p_{\xi_{min}}^c$ and $p_{\xi_{max}}^c$ which all grow like k. If we recall (3.37), (3.27a), (3.27c) and consider the leading order terms then as $\gamma > 1$ it is clear that $p_{\xi_{min}}^c < p_{\xi_{min},\xi_{max}} < p_{\xi_{max}}^c$. Hence the inequality (3.38) holds for $\delta \in [0, 2)$.

Let us now consider the case when $\delta = 2$. If we set $\delta = 2$ in (3.31), (3.36) and (3.37) we see that they all become of order k and hence to establish an inequality we must look at the constants in each. We must therefore compute, for $\delta = 2$,

1. $p_k^c - p_{\xi_{min}}^c$
2. $p_{\xi_{min}}^c - p_{k,\xi_{min}}$
3. $p_{k,\xi_{min}} - p_{k,\xi_{max}}$

4. $p_{k,\xi_{max}} - p_{\xi_{min},\xi_{max}}$

5. $p_{\xi_{min},\xi_{max}} - p^c_{\xi_{max}}$

and show that each is negative.

For brevity we compute only statement (3) and show it is less than zero. The calculations to show that (1)-(5) are less than zero are omitted as the calculations are similar. We know from (3.31) and (3.36) that $p_{k,\xi_{min}} - p_{k,\xi_{max}}$ will be of order k when $\delta = 2$. So instead we compute $(p_{k,\xi_{min}} - p_{k,\xi_{max}})/k$ which is given by,

$$\left(\sqrt{\frac{\frac{5}{2\sqrt{2}} + \frac{1}{2}}{4 - 2\sqrt{2}}} - \left(\frac{\gamma}{\sqrt{2}}\right)^{\frac{1}{2}} \sqrt{\frac{\gamma + \frac{1}{4\gamma^3} - \sqrt{\frac{1}{2}} - \frac{1}{2\sqrt{2}\gamma^2}}{\gamma + \frac{1}{2\gamma} - \sqrt{2}}} \right) = -0.617767\ldots, \text{ when } \gamma = \sqrt{24}. \tag{3.40}$$

Therefore we conclude that $p_{k,\xi_{min}} < p_{k,\xi_{max}}$ is less than zero for $\gamma = \sqrt{24}$ and statement (1) holds. Then as (1)-(5) can all be shown to be less than zero the inequality (3.39) holds. □

In the following subsection we shall calculate the value of $F(\xi, k, \epsilon, p)$ at the intersection points $p_{k,\xi_{min}}, p_{k,\xi_{max}}$, $p_{\xi_{min},\xi_{max}}$ and determine a similar inequality to Corollary 3.14.

3.3.3 Computation of $F(\xi, k, \epsilon, p)$ when $p = p_{k,\xi_{min}}, p_{k,\xi_{max}}$, or $p_{\xi_{min},\xi_{max}}$

We proceed by calculating the value of $F(\xi, k, \epsilon, p)$ at each of these three values found in the previous Lemmas 3.31, 3.36 and 3.37 for k increasing.

In each of the computations that follow we compute $F(\xi, k, \epsilon, p)$ at either one of the relevant ξ for the given intersection in p. For example if we recall the definition of $p_{k,\xi_{max}}$ (3.30b) then this occurs when $F(k, k, \epsilon, p)$ and $F(\xi_{max}, k, \epsilon, p)$ intersect therefore each of these F evaluated with $p_{k,\xi_{max}}$ will be the same. In this example we choose to compute $F(k, k, \epsilon, p_{k,\xi_{max}})$ as it leads to a simpler calculation.

Lemma 3.15:
For k large enough,

$$F(k, k, \epsilon, p_{k,\xi_{max}}) = \frac{k^{\frac{\delta-2}{4}}}{2^{\frac{1}{4}}\sqrt{\gamma}} \left(1 - \mathcal{O}\left(k^{\frac{\delta-2}{4}}\right)\right). \tag{3.41}$$

Proof. We proceed by recalling the definition of $F(\xi, k, \epsilon, p)$, and evaluate with $\xi = k$

and $p = p_{k,\xi_{max}}$

$$F(k,k,\epsilon,p_{k,\xi_{max}}) = \frac{p_{k,\xi_{max}}(\lambda_R(k,k,\epsilon)+\lambda_I(k,k,\epsilon))}{(p_{k,\xi_{max}}+\lambda_R(k,k,\epsilon))^2+(p_{k,\xi_{max}}+\lambda_I(k,k,\epsilon))^2}.$$

Then if we recall the results of Lemma 2.12 it follows that,

$$\begin{aligned} F(k,k,\epsilon,p_{k,\xi_{max}}) &= \frac{p_{k,\xi_{max}}\left(2\sqrt{\frac{\epsilon}{2}}\right)}{2\left(p_{k,\xi_{max}}+\sqrt{\frac{\epsilon}{2}}\right)^2}, \text{ expanding the denominator gives,} \\ &= \frac{p_{k,\xi_{max}}\sqrt{\frac{\epsilon}{2}}}{\left(p_{k,\xi_{max}}^2+2p_{k,\xi_{max}}\sqrt{\frac{\epsilon}{2}}+\frac{\epsilon}{2}\right)}. \end{aligned}$$

Then inserting $\epsilon = k^\delta$ gives,

$$F(k,k,\epsilon,p_{k,\xi_{max}}) = \frac{p_{k,\xi_{max}}\frac{k^{\frac{\delta}{2}}}{\sqrt{2}}}{\left(p_{k,\xi_{max}}^2+2p_{k,\xi_{max}}\frac{k^{\frac{\delta}{2}}}{\sqrt{2}}+\frac{k^\delta}{2}\right)}.$$

We now substitute in the leading order term of (3.31)

$$F(k,k,\epsilon,p_{k,\xi_{max}}) = \frac{\left(\frac{\gamma}{\sqrt{2}}\right)^{\frac{1}{2}}k^{\frac{2+\delta}{4}}\frac{k^{\frac{\delta}{2}}}{\sqrt{2}}}{\left(\left(\frac{\gamma}{\sqrt{2}}\right)k^{\frac{2+\delta}{2}}+\left(\frac{\gamma}{\sqrt{2}}\right)^{\frac{1}{2}}k^{\frac{2+\delta}{4}}\sqrt{2}k^{\frac{\delta}{2}}+\frac{k^\delta}{2}\right)}.$$

Then if we divide the numerator and denominator by a factor of $\left(\frac{\gamma}{\sqrt{2}}\right)k^{\frac{2+\delta}{2}}$ then the above becomes,

$$F(k,k,\epsilon,p_{k,\xi_{max}}) = \frac{k^{\frac{\delta-2}{4}}}{2^{\frac{1}{4}}\sqrt{\gamma}}\left(1+\gamma^{-\frac{1}{2}}2^{\frac{3}{4}}k^{\frac{\delta-2}{4}}+\frac{k^{\frac{\delta-2}{2}}}{\sqrt{2}\gamma}\right)^{-1}. \tag{3.42}$$

For the case $\delta = 2$ then the bracket in (3.42) is just a constant which is clearly positive so the result follows. However if $\delta \in [0,2)$ then if one takes a Taylor expansion of the bracketed term in (3.42) for $k \to \infty$, one can show that,

$$\left(1+\gamma^{-\frac{1}{2}}2^{\frac{3}{4}}k^{\frac{\delta-2}{4}}+\frac{k^{\frac{\delta-2}{2}}}{\gamma\sqrt{2}}\right)^{-1} = 1-\mathcal{O}\left(k^{\frac{\delta-2}{4}}\right), \text{ for } k\to\infty. \tag{3.43}$$

The result then follows. □

As the proofs for $F(\xi,k,\epsilon,p_{k,\xi_{min}})$ and $F(\xi,k,\epsilon,p_{\xi_{min},\xi_{max}})$ are similar, we therefore only present the results.

Lemma 3.16:
For k large enough,

$$F(k,k,\epsilon,p_{k,\xi_{min}}) = \frac{k^{\frac{\delta-2}{4}}}{2^{\frac{1}{4}}}\left(1-\mathcal{O}\left(k^{\frac{\delta-2}{2}}\right)\right). \tag{3.44}$$

Lemma 3.17:
For k large enough,

$$F(\xi_{min},k,\epsilon,p_{\xi_{min},\xi_{max}}) = \frac{\sqrt{\gamma}}{2^{\frac{1}{2}}\beta^1}\left(1-\mathcal{O}\left(k^{\delta-2}\right)\right), \tag{3.45}$$

where $\beta = \gamma + 1 + \sqrt{2\gamma}$.

We can then use Lemmas 3.15, 3.16 and 3.17 to immediately prove the following result,

Theorem 3.18:
If $\epsilon = ck^{\delta}$ and $\xi \in \{\xi_{min}, k, \xi_{max}\}$, then for k sufficiently large and $\delta \in [0,2)$ it follows that,

$$F(\xi,k,\epsilon,p_{k,\xi_{max}}) < F(\xi,k,\epsilon,p_{k,\xi_{min}}) < F(\xi,k,\epsilon,p_{\xi_{min},\xi_{max}}). \tag{3.46}$$

And for k sufficiently large and $\delta = 2$ it follows that,

$$F(\xi,k,\epsilon,p_{k,\xi_{max}}) < F(\xi,k,\epsilon,p_{\xi_{min},\xi_{max}}) < F(\xi,k,\epsilon,p_{k,\xi_{min}}). \tag{3.47}$$

Proof. If we consider the case for $\delta \in [0,2)$ first then we need only take the leading order terms of (3.41), (3.44) and (3.45). To start we will compare (3.41) and (3.44). On doing this one can observe that immediately that

$$F(k,k,\epsilon,p_{k,\xi_{max}}) = \gamma^{-\frac{1}{2}} F(k,k,\epsilon,p_{k,\xi_{min}}).$$

Then as $\gamma^{-\frac{1}{2}} < 1$ it follows that,

$$F(k,k,\epsilon,p_{k,\xi_{max}}) < F(k,k,\epsilon,p_{k,\xi_{min}}), \text{ for } k \text{ sufficiently large enough.}$$

We now show that $F(k,k,\epsilon,p_{k,\xi_{min}}) < F(k,k,\epsilon,p_{\xi_{min},\xi_{max}})$, by computing the difference between $F(k,k,\epsilon,p_{k,\xi_{min}})$ and $F(k,k,\epsilon,p_{\xi_{min},\xi_{max}})$,

$$\begin{aligned} F(k,k,\epsilon,p_{k,\xi_{min}}) - F(k,k,\epsilon,p_{\xi_{min},\xi_{max}}) &= \frac{k^{\frac{\delta-2}{4}}}{2^{\frac{1}{4}}} - \frac{\sqrt{\gamma}}{2^{\frac{1}{2}}\beta^1}, \text{ dividing by } \frac{\sqrt{\gamma}}{2^{\frac{1}{2}}\beta^1}, \\ &= \frac{2^{\frac{1}{4}}\beta k^{\frac{\delta-2}{4}}}{\sqrt{\gamma}} - 1 \end{aligned}$$

Then for k sufficiently large enough $\frac{2^{\frac{1}{4}}\beta k^{\frac{\delta-2}{4}}}{\sqrt{\gamma}} \to 0$, and therefore,

$$F(k,k,\epsilon,p_{k,\xi_{min}}) - F(k,k,\epsilon,p_{\xi_{min},\xi_{max}}) < 0$$

Therefore we have the that, for $\delta \in [0,2)$ and $\xi \in \{\xi_{min}, k, \xi_{max}\}$

$$F(k,k,\epsilon,p_{k,\xi_{max}}) < F(\xi,k,\epsilon,p_{k,\xi_{min}}) < F(\xi,k,\epsilon,p_{\xi_{min},\xi_{max}}),$$

for k sufficiently large enough.

We now consider the case when $\delta = 2$ and prove that (3.47) holds. Firstly we recall (3.42), and let $\delta = 2$ to attain.

$$F(k,k,\epsilon,p_{k,\xi_{max}}) = \frac{\sqrt{\gamma}}{2^{\frac{1}{4}}}\left(\gamma + \sqrt{\gamma}2^{\frac{3}{4}} + \sqrt{\frac{1}{2}}\right)^{-1}. \tag{3.48}$$

Similarly we can compute similarly expressions using (3.44) and (3.45) and setting $\delta = 2$,

$$F\left(\xi_{min},k,\epsilon,p_{\xi_{min},\xi_{max}}\right) = \sqrt{\frac{9\gamma}{8}}\left(\frac{5}{4} + \gamma + 3\sqrt{\frac{\gamma}{2}}\right)^{-1} \tag{3.49}$$

$$F\left(k,k,\epsilon,p_{\xi_{min},k}\right) = \frac{2^{\frac{1}{4}}}{\sqrt{2}+2^{\frac{5}{4}}+1}. \tag{3.50}$$

As mentioned previously in our numerical computations we choose $\gamma = \sqrt{24}$. Therefore one can directly compute the difference of (3.48) and (3.49) and (3.49) and (3.50) where,

$$\begin{aligned} F(k,k,\epsilon,p_{k,\xi_{max}}) - F\left(\xi_{min},k,\epsilon,p_{\xi_{min},\xi_{max}}\right) &= -0.016968\ldots, \\ F\left(\xi_{min},k,\epsilon,p_{\xi_{min},\xi_{max}}\right) - F\left(k,k,\epsilon,p_{\xi_{min},k}\right) &= -0.031646\ldots. \end{aligned}$$

Therefore when $\gamma = \sqrt{24}$ then the inequality (3.47) holds. □

3.4 Proof of the main results

We now use the results of the previous section to prove the main results of this chapter, namely Theorem 3.2 and the Corollary 3.3.

Proof of Theorem 3.2. Consider the graphs of $F(\xi_{max}, k, \epsilon, p)$ and $F(k, k, \epsilon, p)$ as functions of p. These graphs have got one intersection point $p = p_{k,\xi_{max}}$ and the configuration of the graphs is shown in Figures 3-5, 3-6, 3-7 and 3-8. So then the minimum of these two functions is the envelope below the point of intersection $p = p_{k,\xi_{max}}$, which is maximised at the value $F(k, k, \epsilon, p_{k,\xi_{max}})$. This can be proved by the lemmas above. Similarly one can show that the maximum of the minimum of the two curves $F(\xi, k, \epsilon, p)$ for $\xi = \xi_{min}$ and k is at $F(k, k, \epsilon, p_{\xi_{min},k})$. Moreover the two curves with $\xi = \xi_{min}$ and $\xi = \xi_{max}$ have their minimum maximised at the point with $F(\xi_{min}, k, \epsilon, p_{\xi_{min},\xi_{max}})$. Now taking the minimum of the three envelopes and using (3.46) to compares the three relevant values of F we see that

$$\max_{p\in\mathbb{R}^+}\left(\min_{\xi_{min}\leq\xi\leq\xi_{max}} F(\xi, k, \epsilon, p)\right) = F(k, k, \epsilon, p_{k,\xi_{max}}).$$

Then (3.11) follows from (3.31). □

In Figures 3-5, 3-6, 3-7 and 3-8 we plot $F(\xi_{min}, k, \epsilon, p)$, $F(k, k, \epsilon, p)$, and $F(\xi_{max}, k, \epsilon, p)$ for a given k, ϵ, ξ_{min} and ξ_{max}. In each figure the solution of (3.6) can be seen to be $p_{k,\xi_{max}}$.

We can now use the solution to (3.9), $p_{k,\xi_{max}}$ to compute $\rho_0^O(\xi, k, \epsilon, p_{k,\xi_{max}}, 0)$ and evaluate when k increases. If we then recall the result of Lemma 3.11, namely equation (3.31), then if we let $\gamma = \sqrt{24}$ in this equation then we obtain the advertised result in equation (3.11).

Proof of Corollary 3.3. If we recall the definition of the convergence rate (3.7),

$$\rho_0^O(\xi, k, \epsilon, p, 0) = 1 - 4F(\xi, k, \epsilon, p).$$

We therefore wish to evaluate $F(\xi, k, \epsilon, p)$ with the optimised choice of $p^* = p_{k,\xi_{max}}$, and then find the maximum of $\rho_0^O(\xi, k, \epsilon, p_{k,\xi_{max}}, 0)$ (and hence the minimum of $F(\xi, k, \epsilon, p)$) which we expect to occur at $\xi = \xi_{max}$. This would give us a conservative estimate of the convergence rate, as k increases. If we recall the result of Lemma 3.15 then,

$$F(\xi_{max}, k, \epsilon, p_{k,\xi_{max}}) = \frac{k^{\frac{\delta-1}{4}}}{2^{-\frac{1}{4}}\sqrt{\gamma}}\left(1 - \mathcal{O}\left(k^{\frac{\delta-1}{4}}\right)\right). \tag{3.51}$$

for k increasing. If one then inserts (3.51) into (3.7) the result follows. □

We can now use this result to prove Corollary 3.4 and provide a lower bound on the number of iterations of (2.17) needed to reach a tolerance τ.

Proof of Corollary 3.4. If we recall (2.21) then one can see that the error of the Schwarz algorithm (2.17) at iterate two (after one solve on each subdomain) is given by

$$\widehat{E^2}(x,\xi) = \rho_0^O\left(\xi, k, \epsilon, p^*, 0\right)\widehat{E^0}(x,\xi).$$

If we now take the L^2 norm of both sides it is true that

$$\left\|\widehat{E^2}(x,\xi)\right\|_{L^2} \leq \left|\rho_0^O\left(\xi, k, \epsilon, p^*, 0\right)\right| \left\|\widehat{E^0}(x,\xi)\right\|_{L^2},$$

and hence after N_{iters} iterates it is true that,

$$\left\|\widehat{E^{N_{iters}}}(x,\xi)\right\|_{L^2} \leq \left|\rho_0^O\left(\xi, k, \epsilon, p^*, 0\right)\right|^{2N_{iters}} \left\|\widehat{E^0}(x,\xi)\right\|_{L^2}.$$

Then as $\|\hat{E}(x,\xi)\|_{L^2} = \|E(x,y)\|_{L^2}$ by Plancherel's Theorem it follows that

$$\left\|E^{N_{iters}}(x,\xi)\right\|_{L^2} \leq \left|\rho_0^O\left(\xi, k, \epsilon, p^*, 0\right)\right|^{2N_{iters}} \left\|E^0(x,\xi)\right\|_{L^2}.$$

As $\rho_0^O < 1$ by Theorem 3.1 then it is certainly true that $\frac{\left\|E^{N_{iters}}\right\|_{L^2}}{\|E^0\|_{L^2}} \to 0$ as $N_{iters} \to \infty$. But of course in practice we want our Schwarz algorithm to converge to some allowable tolerance τ (say 10^{-6}) in a finite number of iterations N_{iters}. Hence we rewrite the above as,

$$\begin{aligned}\frac{\left\|E^{2N_{iters}}(x,y)\right\|_{L^2}}{\|E^0(x,y)\|_{L^2}} &\leq \left|\rho_0^O\left(\xi, k, \epsilon, p^*, 0\right)\right|^{N_{iters}}, \\ &\leq \tau.\end{aligned}$$

As we already have an expression for the maximum value of $|\rho_0^O|$ with respect to ξ for large k then we can use the above equation. This will allow us to show how the number of iterations taken for the Schwarz algorithm (2.17) to reach τ depends on k. We start by writing down the following inequality using the equation above,

$$\left|\rho_0^O\left(\xi, k, \epsilon, p^*, 0\right)\right|^{2N_{iters}} \leq \tau.$$

If one then takes the log of both sides this yields

$$-2N_{iters}\log\left(\left|\rho_0^O\left(\xi,k,\epsilon,p^*,0\right)\right|\right)\geq-\log\left(\tau\right).$$

Then if we recall that the leading order terms of $\max_\xi|\rho_0^O|$ are given by (3.12), and substitute into the above equation this gives

$$-2N_{iters}\log\left(1-\frac{4}{2^{\frac{1}{4}}\sqrt{\gamma}}k^{\frac{\delta-2}{4}}\right)\geq-\log\left(\tau\right).$$

Then performing a series expansion of the log term on the left hand side of the above equation for large k yields

$$\begin{aligned}N_{iters}\frac{8}{2^{\frac{1}{4}}\sqrt{\gamma}}k^{\frac{\delta-2}{4}}&\geq-\log\left(\tau\right),\\&=\log\left(\frac{1}{\tau}\right)\end{aligned}$$

If we then rearrange the above, and substitute $\gamma=\sqrt{24}$ we get the result (3.13),

$$N_{iters}\geq\log\left(\frac{1}{\tau}\right)\frac{3^{\frac{1}{4}}}{4}k^{\frac{2-\delta}{4}}$$

□

In the following Chapter we will implement the Schwarz algorithm and test all of the interface conditions mentioned in Chapters 2 and 3. These numerical computations will verify that indeed the optimised interface condition improves significantly on the standard Taylor conditions as k increases, a result which Corollaries 3.3 and 3.4 would suggest.

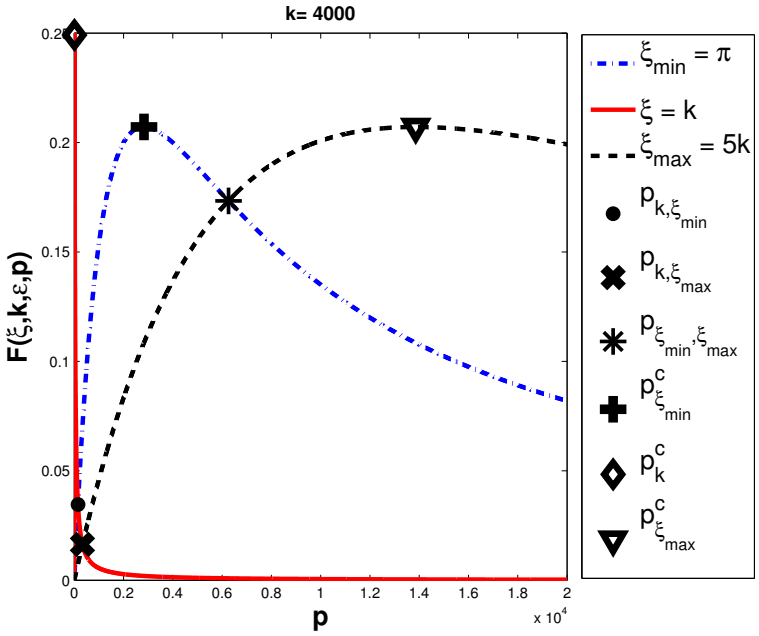

Figure 3-5: *Plot of the three functions* $F(\xi_{min}, k, \epsilon, p)$, $F(k, k, \epsilon, p)$ *and* $F(\xi_{max}, k, \epsilon, p)$ *with* $k = 4000$, $\epsilon = k^{\frac{1}{2}}$, $\xi_{min} = \pi$ *and* $\xi_{max} = 5k$.

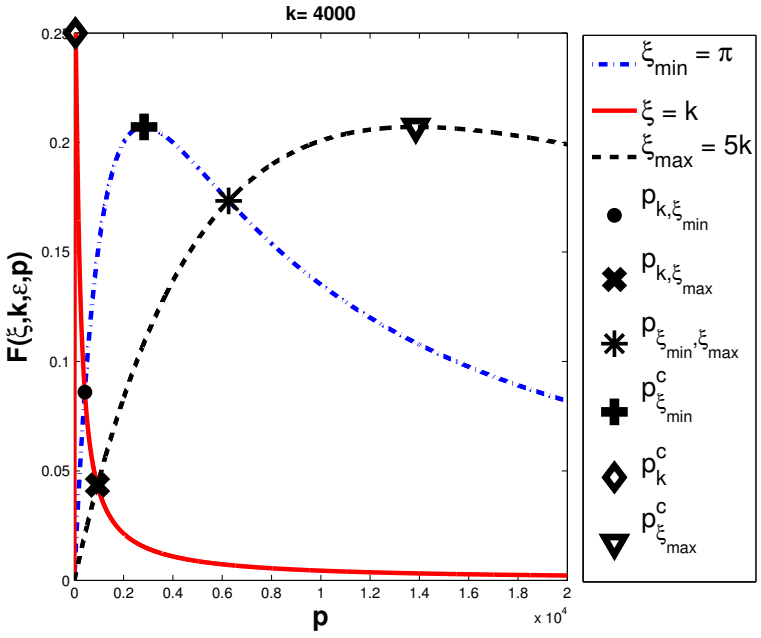

Figure 3-6: *Plot of the three functions* $F(\xi_{min}, k, \epsilon, p)$, $F(k, k, \epsilon, p)$ *and* $F(\xi_{max}, k, \epsilon, p)$ *with* $k = 4000$, $\epsilon = k$, $\xi_{min} = \pi$ *and* $\xi_{max} = 5k$.

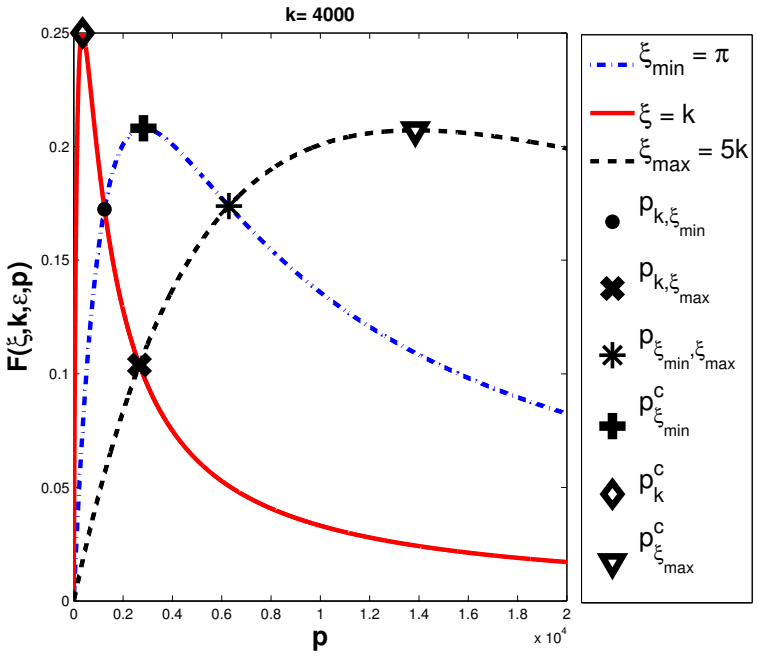

Figure 3-7: *Plot of the three functions* $F(\xi_{min}, k, \epsilon, p)$, $F(k, k, \epsilon, p)$ *and* $F(\xi_{max}, k, \epsilon, p)$ *with* $k = 4000$, $\epsilon = k^{\frac{3}{2}}$, $\xi_{min} = \pi$ *and* $\xi_{max} = 5k$.

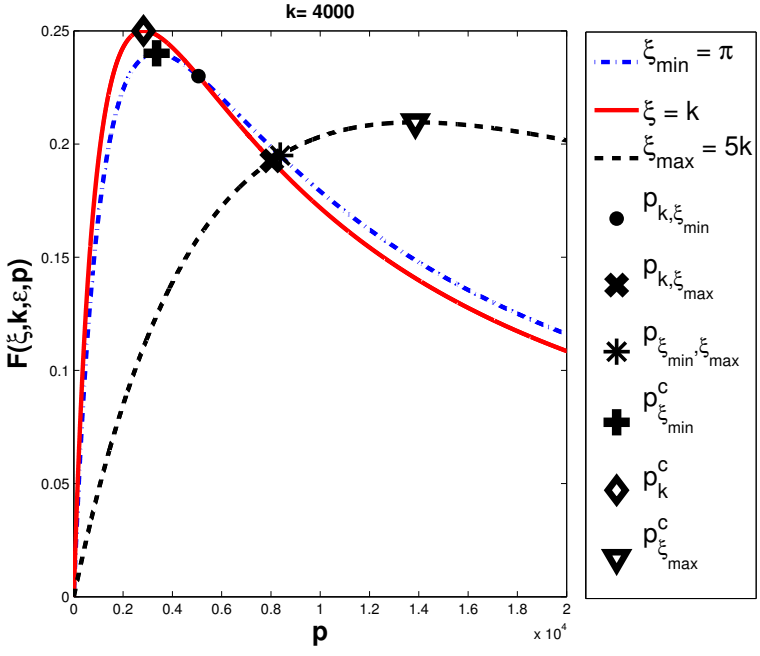

Figure 3-8: *Plot of the three functions* $F(\xi_{min}, k, \epsilon, p)$, $F(k, k, \epsilon, p)$ *and* $F(\xi_{max}, k, \epsilon, p)$ *with* $k = 4000$, $\epsilon = k^2$, $\xi_{min} = \pi$ *and* $\xi_{max} = 5k$.

CHAPTER 4

NUMERICAL EXPERIMENTS WITH SCHWARZ DOMAIN DECOMPOSITION METHODS

In this chapter we perform two types of numerical experiments involving the Schwarz algorithm (2.17). The overall aim of these experiments is to illustrate the effect of different choices of the interface operator S.

The first set of numerical experiments is concerned with the actual construction of S and its effect on the convergence rate ρ. Considering first the case

$$Su = p\,(1+i)\,u, \text{ with } p > 0, \tag{4.1}$$

we recall that in Chapter 3 we solved the minimax problem (3.6) to obtain a closed form solution p^* to the minimax problem and examined its asymptotics as $k \to \infty$. The resulting convergence rate is given by (3.3), derived by considering a two domain non-overlapping Schwarz iterative method. In this chapter we consider first the addition of overlap into the Schwarz algorithm (2.17) in the case when the interface operator S is of the form (4.1), and how this effects the solution of the minimax problem and the resulting convergence rate. Secondly we consider optimising over two free parameters by the addition of a higher order term in the optimised interface condition in the form of a second order tangential derivative. That is

$$Su = p\,(1+i)\,u - q\,(1+i)\,\frac{\partial^2 u}{\partial y^2}, \text{ with } p, q > 0, \tag{4.2}$$

If we recall the form of the standard second order Taylor interface condition (2.55) then it is clear why we do not include any terms in $\frac{\partial u}{\partial y}$ In both cases we solve the resulting minimax problems numerically with either one of these new additions, and

conjecture using the numerical results as to how the solutions depend on k and the absorbing parameter ϵ as k increases. This will then allow us to conjecture as to how the maximum of the convergence rate of the Schwarz algorithm behaves as k increases. What we show is that both of these additions can provide an improvement in the convergence of the Schwarz iteration compared to the case with no overlap and interface condition (4.1), see §4.1.1 and §4.1.2.

The second set of numerical tests are a study of the Schwarz iterative algorithm for a two subdomain decomposition (2.17). We use this method as both an iterative method and as a preconditioner for GMRES, using methods discussed in chapters 2, 3 and the first section of this chapter. These results provide numerical evidence to the theoretical bounds provided in the chapters mentioned.

In the final set of numerical experiments we test the influence of the choice of the interface condition in a situation where we no longer have any theoretical guidance for the convergence of the method. In these numerical experiments we use a one iteration of a restricted additive Schwarz method (RAS) as a preconditioner for GMRES, and our domain is decomposed into two or more subdomains. What these experiments show is that in this more general scenario we no longer observe as much of a difference between the optimised and Taylor interface conditions, but in general the optimised method still converges quicker.

4.1 Numerical solution of the minimax problem

In this section we shall consider numerically solving minimax problems of a similar form to (3.5) using our own subroutine which uses the Nelder-Mead simplex method [42] to solve the minimisation part of the minimax problem. We then use these numerical results to conjecture how the optimised parameters and convergence rate behave as k increases. To motivate why we consider solving the resulting minimax problems numerically let us consider the case of the overlapping Schwarz iterative method (given by (2.17) with $L > 0$) with zeroth order impedance transmission condition (3.2). One can then show that the convergence rate of this iterative method is given by,

$$\left|\rho_0^O(\xi,k,\epsilon,p,L)\right| = \frac{\left(p-\lambda_R(\xi,k,\epsilon)\right)^2+\left(p-\lambda_I(\xi,k,\epsilon)\right)^2}{\left(p+\lambda_R(\xi,k,\epsilon)\right)^2+\left(p+\lambda_I(\xi,k,\epsilon)\right)^2}e^{-2L\lambda_R(\xi,k,\epsilon)}. \tag{4.3}$$

Then the resulting minimax problem which we solve is given by,

$$\min_{p\in\mathbb{R}_+}\left(\max_{\xi_{min}\leq\xi\leq\xi_{max}}\left|\rho_0^O(\xi,k,\epsilon,p,L)\right|\right), \tag{4.4}$$

Previously we solved (3.5) in Chapter 3 and obtained information about the behaviour of the convergence rate (3.3) with respect to ξ and p to enable us to solve the minimax problem. Solving the minimax problem (4.4) in more generality analytically seems to be very technically demanding. Thus instead we choose to solve the minimax problem (4.4) numerically and conjecture as to how the leading asymptotic terms of the solution p behaves with k. Therefore we use the Nelder-Mead method as it is a derivative free non-linear optimisation method. In the numerical implementations in the second section of this chapter we use the values for optimised parameters generated by the numerical solution of (4.4).

4.1.1 Zeroth order interface condition with overlap

We start by solving (4.4) numerically and then conjecture what the leading order asymptotic behaviour of p is, for increasing k. We are interested in the asymptotic behaviour as the end goal is to have an efficient solver when solving Helmholtz problems with large values of k. Therefore it is useful for us to know how the convergence rate ρ behaves for increasing k. For the following results we assume that $h = \frac{\pi}{5k}$ and that our overlap is a fixed number of grid points and so $L \sim h$. In Figure 4-2 the solution, p, of (4.4) is plotted for a fixed ϵ and increasing k. From these numerical experiments we then make the following conjecture of the asymptotic behaviour of p with respect to k.

Conjecture 4.1:
Assuming that $\epsilon = k^{\delta}$, where $\delta \in [0, 2]$, $\xi_{min} \geq 0$ (and independent of k), $\xi_{max} = 5k$, and an overlap $L \sim k^{-1}$ (as we assume that $h \sim k^{-1}$). Then the p which solves (4.4) behaves like the following for increasing k,

$$p \sim k^{\frac{\delta+1}{3}}. \tag{4.5}$$

This conjecture then allows us to comment on the behaviour of the maximum of the convergence rate (4.3) for increasing k. We present the following conjecture describing the leading order asymptotic behaviour of the maximum (4.3) for k increasing and then justify this numerically.

Corollary 4.2:
Assuming that Conjecture 4.1 holds where p is the solution of (4.4), then the maximum of the convergence rate (4.3) behaves as follows for k increasing,

$$\max_{\xi_{min} \leq \xi \leq \xi_{max}} \left| \rho_0^O(\xi, k, \epsilon, p, L) \right| \sim 1 - k^{\frac{\delta-2}{6}} + \mathcal{O}\left(k^{\frac{\delta-2}{3}}\right). \tag{4.6}$$

In Figure 4-1 we plot the convergence rate (4.3) for $k = 1000$, $\epsilon = k$, $L = \frac{\pi}{5k}$ and $\xi \in [0, 5k]$. What we observe is that the convergence rate has two internal maxima, where the greater of the two maxima occurs at $\xi = k$. We then conjecture from this that,

$$\begin{aligned}\max_{\xi_{min}\leq\xi\leq\xi_{max}}\left|\rho_0^O(\xi,k,\epsilon,p,L)\right| &= \left|\rho_0^O(k,k,\epsilon,p,L)\right|, \\ &= \frac{(p-\lambda_R(k,k,\epsilon))^2+(p-\lambda_I(k,k,\epsilon))^2}{(p+\lambda_R(k,k,\epsilon))^2+(p+\lambda_I(k,k,\epsilon))^2}e^{-2L\lambda_R(k,k,\epsilon)}.\end{aligned} \tag{4.7}$$

We can then substitute (4.5) for p into (4.7), and then using the fact that $\lambda_R(k,k,\epsilon) = \lambda_I(k,k,\epsilon) = \sqrt{\frac{k^\delta}{2}}$ and that $L \sim k^{-1}$ this tells us that,

$$\max_{\xi_{min}\leq\xi\leq\xi_{max}}\left|\rho_0^O(\xi,k,\epsilon,p,L)\right| \sim \frac{\left(k^{\frac{\delta+1}{3}}-\sqrt{\frac{k^\delta}{2}}\right)^2}{\left(k^{\frac{\delta+1}{3}}+\sqrt{\frac{k^\delta}{2}}\right)^2}e^{-2k^{\frac{\delta-2}{2}}}.$$

If we then Taylor expand the right hand side of the above the equation for large k then we obtain the desired result (4.6).

In Figure 4-3 we plot the convergence rate (4.3) for increasing k with $\xi_{min} = \pi$, $\xi_{max} = 5k$, $L = \frac{\pi}{5k}$ and fixed ϵ. The choice of p that we use is that generated by numerically solving (4.4) with the parameters given above. What we observe is that the numerically computed value of the maximum of the convergence rate agrees quite well with the asymptotic behaviour predicted by Corollary 4.2 especially as k increases.

We recall the analogous result with no overlap, given in Corollary 3.3, tells us that $\rho \sim 1 - k^{\frac{\delta-2}{4}}$. Therefore the addition of an overlap results in a convergence rate which does not deteriorate as badly when k increases.

Remark 4.3:
We note that the estimate given in (4.6) *assumes that* $L = C_L h$ *for some* $C_L > 0$*, and thus* C_L *affects the hidden constant in the asymptotic expression* (4.5)*. This dependence is somehow crucial as when* $C_L = 0$ *we should recover the result without overlap* (3.12) *which is worse. Hence further work could be done to find the explicit dependence of the hidden constant in* (4.5) *on* C_L*. However this is outside the scope of this thesis.*

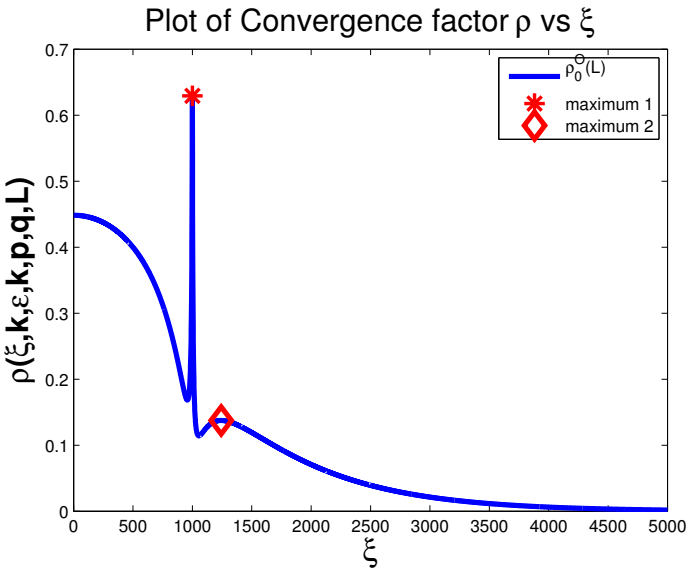

Figure 4-1: *We plot the convergence rate* (4.3) *with* p *given by the solution of* (4.4) *for increasing* ξ*. The internal maxima are indicated by the asterisk and diamond. We fix* $k = 1000$*,* $\epsilon = k$*,* $\xi_{max} = 5k$ *and* $L = h$*.*

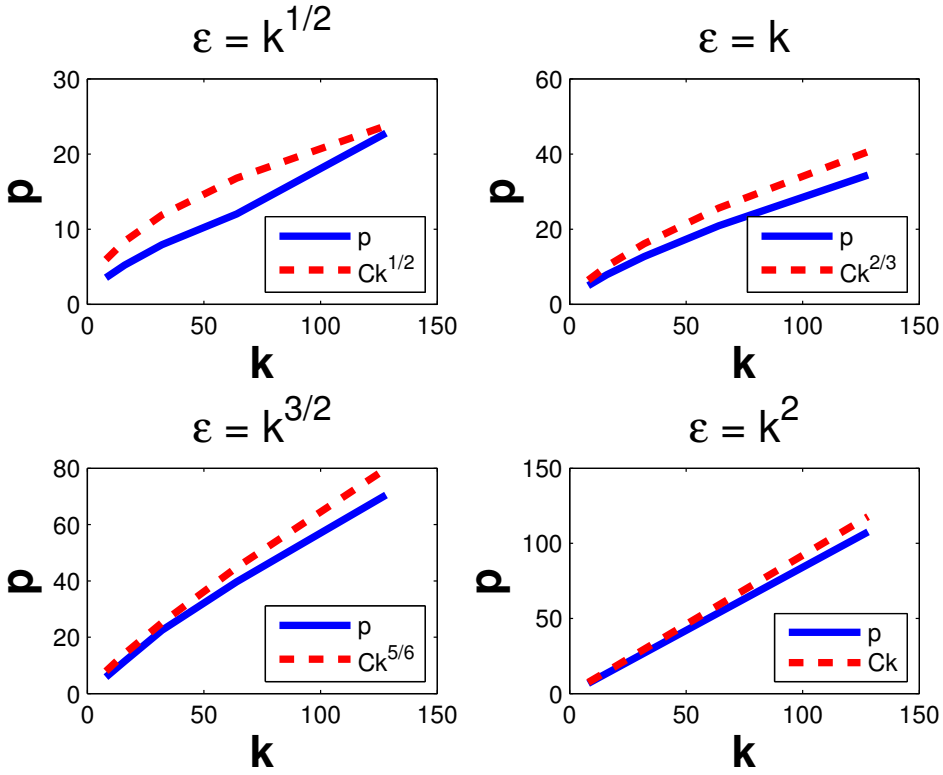

Figure 4-2: *For each Figure we plot the numerical solution* p *of* (4.4) *vs* k *with a given value of* ϵ*. We fix* $\xi_{min} = \pi$*,* $\xi_{max} = 5k$ *and* $L = h$*.*

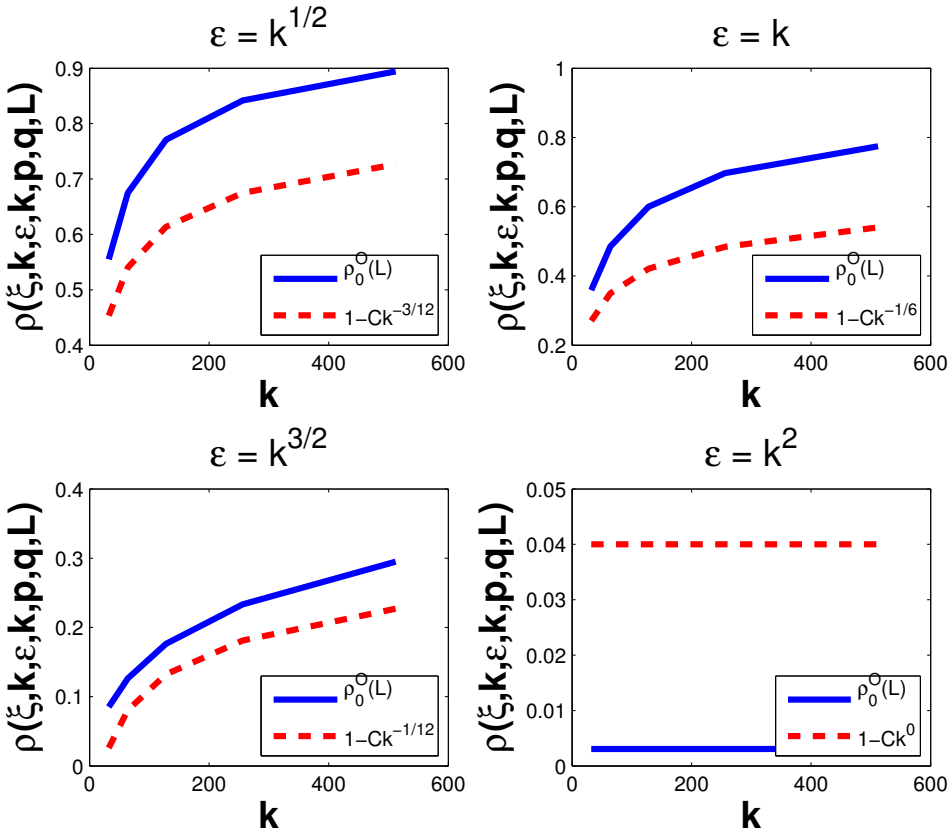

Figure 4-3: *For each Figure we plot the convergence rate ρ vs k with a given value of ϵ and p given by the solution of* (4.4)*. We fix $\xi_{min} = \pi$, $\xi_{max} = 5k$ and $L = h$.*

4.1.2 Second order interface condition without overlap

In section 2.4.1 of Chapter 2 we discussed the addition of a second order term into the Taylor interface condition (2.54). We then proved in Theorems 2.29 and 2.30 that the convergence rate is better in this case compared to the case when there is a zero order term. Therefore we now include a second order term in the optimised method in the hope that optimising with this additional parameter will give improved convergence. Hence we consider now interface operators of the form,

$$S_2^O = p(1+i) - q(1+i)\partial_{yy}^2, \quad \text{and thus,} \tag{4.8}$$

and thus,

$$\sigma_2^O = p(1+i) + q(1+i)\xi^2,$$

where p, q are to be chosen. Therefore in our optimisation problem we now have two parameters p, q which we can use to minimise the maximum of the convergence rate. The minimax problem which we consider solving is the following,

$$\min_{p,q\in\mathbb{R}_+} \left(\max_{\xi_{min}\leq\xi\leq\xi_{max}} \left| \rho_2^O(\xi, k, \epsilon, p, q, 0) \right| \right), \tag{4.9}$$

where,

$$\left|\rho_2^O(\xi, k, \epsilon, p, q, 0)\right| = \frac{\left(p + q\xi^2 - \lambda_R(\xi, k, \epsilon)\right)^2 + \left(p + q\xi^2 - \lambda_I(\xi, k, \epsilon)\right)^2}{\left(p + q\xi^2 + \lambda_R(\xi, k, \epsilon)\right)^2 + \left(p + q\xi^2 + \lambda_I(\xi, k, \epsilon)\right)^2}. \tag{4.10}$$

As we have done previously, we solve (4.9) numerically then conjecture as to what the leading asymptotic behaviour of p and q are for increasing k, assuming that $h = \frac{\pi}{5k}$. In Figures 4-4 and 4-5 the solutions p (left of figures) and $\log(q)$ (right of Figure) of (4.4) are plotted for a fixed ϵ and increasing k. We plot $\log(q)$ instead of q as this makes the asymptotic behaviour of q with respect to k clearer. From these numerical results we make the following conjecture.

Conjecture 4.4:
Assuming that $\epsilon = k^\delta$, where $\delta \in [0, 2]$, $\xi_{min} \geq 0$ (and independent of k), $\xi_{max} = 5k$, and no overlap. Then the p and q which solve (4.9) *behave like the following for increasing k,*

$$p \sim k^{\frac{\delta+2}{4}}, \tag{4.11}$$

$$q \sim k^{\frac{\delta-6}{4}}. \tag{4.12}$$

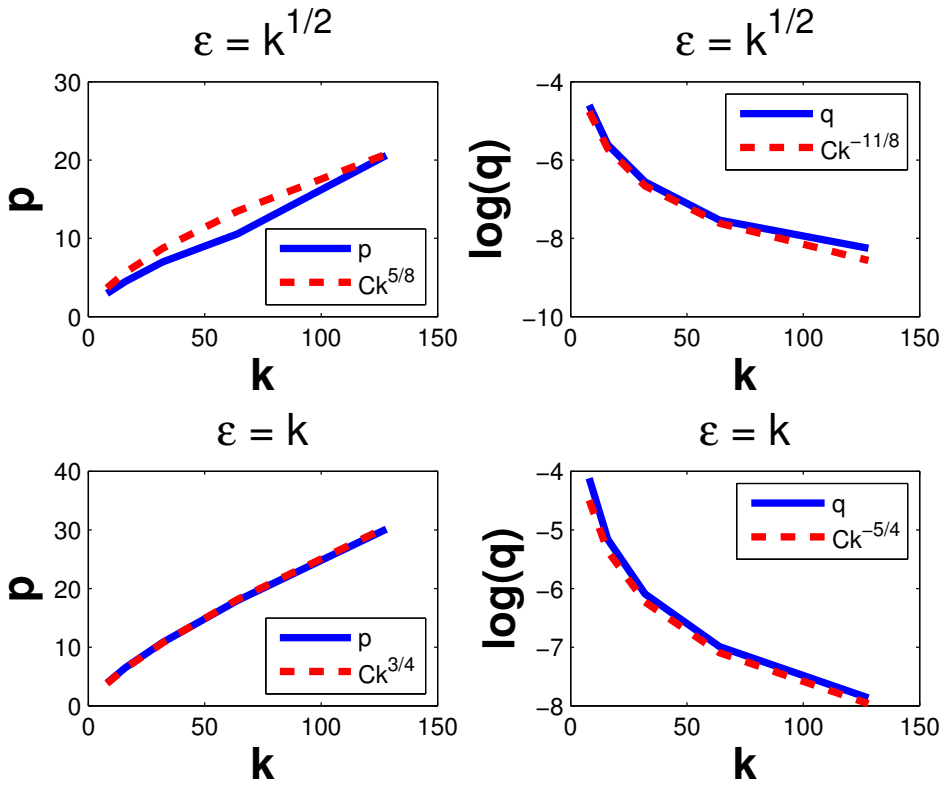

Figure 4-4: *For each Figure we plot the numerical solutions p and q of (4.9) vs k with a given value of ϵ. We fix $\xi_{min} = \pi$, $\xi_{max} = 5k$.*

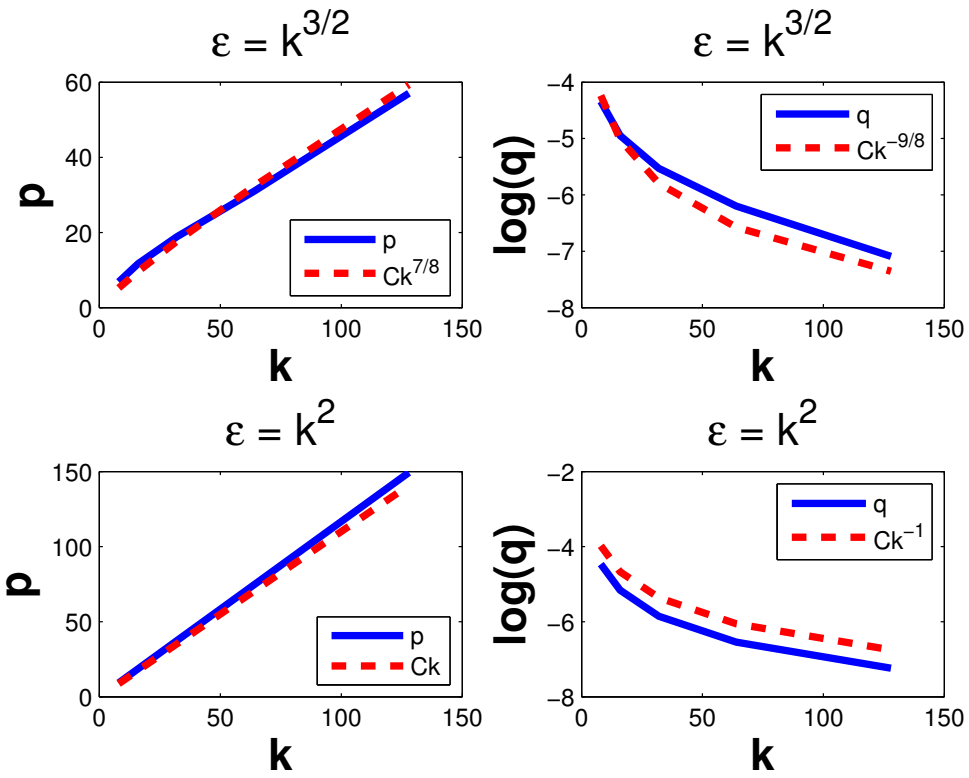

Figure 4-5: *For each Figure we plot the numerical solutions p and q of (4.9) vs k with a given value of ϵ. We fix $\xi_{min} = \pi$, $\xi_{max} = 5k$.*

We now evaluate the maximum of the convergence rate (4.10) numerically with the values of p and q found by numerically solving (4.9). The results of this for increasing k with $\xi_{min} = \pi$, $\xi_{max} = 5k$, $h = \frac{\pi}{5k}$ and fixed ϵ can be found in Figure 4-6. From these numerical results we make the following observation as to the behaviour of the maximum with respect to ξ of (4.10).

Corollary 4.5:
Under the same assumptions as Conjecture 4.4, and if p and q are the solutions obtained from (4.9)*, then the maximum of the convergence rate* (4.10) *behaves like the following for k increasing,*

$$\max_{\xi_{min} \leq \xi \leq \xi_{max}} \left| \rho_2^O(\xi, k, \epsilon, p, q, 0) \right| \sim 1 - k^{\frac{\delta-2}{8}} + \mathcal{O}\left(k^{\frac{\delta-2}{4}}\right). \tag{4.13}$$

What we observe is that with the addition of an extra free parameter q, which comes from a second order tangential derivative in the interface condition, we should expect to achieve even better convergence of the iterative Schwarz algorithm (2.17) compared to all of the choices mentioned previously. For example in Corollary 4.2 we observed an asymptotic rate of $\rho \sim 1 - k^{\frac{\delta-2}{6}}$. Therefore we should expect to observe a slower growth in the number of iterations of the iterative Schwarz method (2.17) for increasing k when we use the optimised second order conditions. In Figure 4-6 we plot (in blue) the actual value of the the maximum of $|\rho_2^O(\xi, k, \epsilon, p, q, 0)|$ for increasing k with a fixed ϵ with p and q computed numerically by solving (4.9). This is compared with the asymptotic result from conjecture 4.5 (given in red). We observe that when k increases the rate of growth of the maximum computed numerically agrees well with the behaviour of the asymptotic result. Hence we can conclude that the numerical evidence suggests that the conjectured asymptotics in (4.13) are indeed correct.

In the next section we shall implement the iterative Schwarz algorithm (2.17) with all of the interface conditions which we have discussed in chapters 2 and 3 and this section. We shall use these results to provide further evidence for the convergence rate bounds shown previously for both optimised methods and the Taylor methods.

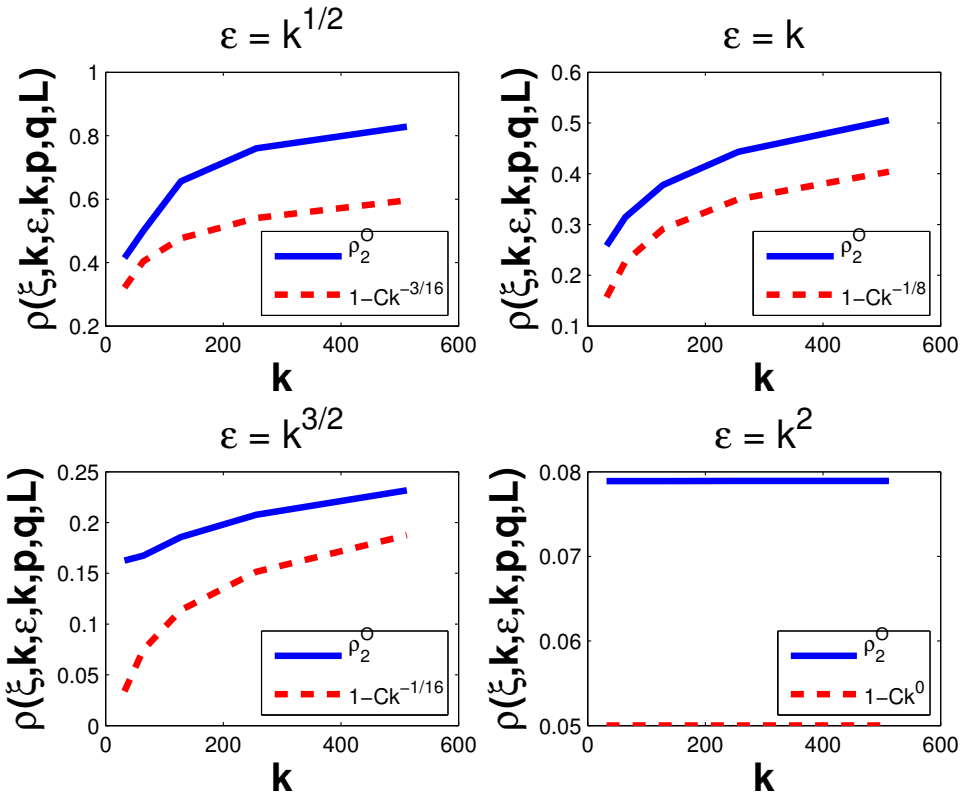

Figure 4-6: *For each Figure we plot the convergence rate ρ vs k with a given value of ϵ and p and q given by the solutions of (4.9). We fix $\xi_{min} = \pi$, $\xi_{max} = 5k$ and $L = h$.*

4.2 Numerical experiments using the Schwarz method on 2 subdomains

We now demonstrate the convergence of the Schwarz method and compare the various interface conditions mentioned previously. First let us consider the following Helmholtz problem on the unit square with an impedance boundary condition,

$$\left.\begin{aligned} -\Delta u - k^2 u + i\epsilon u = 1, \text{ in } \Omega = (0,1)^2 \\ \frac{\partial}{\partial n} u + iku = 0, \text{ on } \partial\Omega \end{aligned}\right\} \tag{4.14}$$

where we remind the reader that $k, \epsilon > 0$ and $\frac{\partial}{\partial n}$ denotes the outward normal derivative. We then choose to discretise (4.14) with piecewise linear finite elements on a uniform

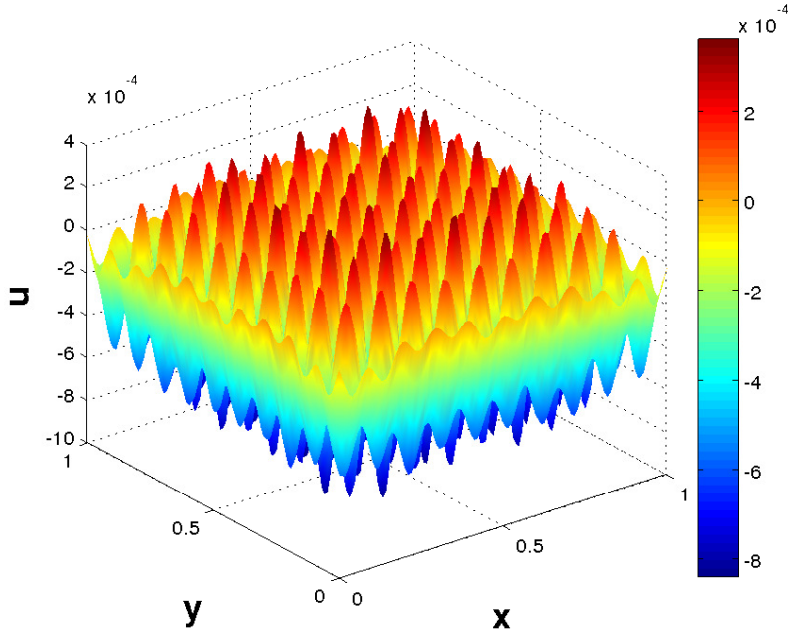

Figure 4-7: *Numerical solution of* (4.14) *where* $k = 20\pi$ *and* $\epsilon = k$.

grid with grid spacing h to obtain the following linear system,

$$A_\epsilon \mathbf{U} = \mathbf{1}. \tag{4.15}$$

The numerical solution of the above PDE is shown in Figure 4-7 for a fixed k and ϵ. We now choose to either solve (4.15) using the iterative Schwarz method (2.17), or use one iteration of (2.17) to approximate A_ϵ^{-1} (recall that we denote this approximate inverse

by M_ϵ^{-1}) and solve the following preconditioned linear system with GMRES,

$$M_\epsilon^{-1} A_\epsilon \mathbf{U} = M_\epsilon^{-1} \mathbf{1}. \tag{4.16}$$

For the Schwarz method we decompose Ω into two subdomains which are allowed an overlap of L each,

$$\Omega_1 = \left(0, \frac{1}{2} + L\right) \times (0, 1)\,,$$
$$\Omega_2 = \left(\frac{1}{2} - L, 1\right) \times (0, 1)\,.$$

We solve the resulting subdomain problems using a direct solver. For each iteration

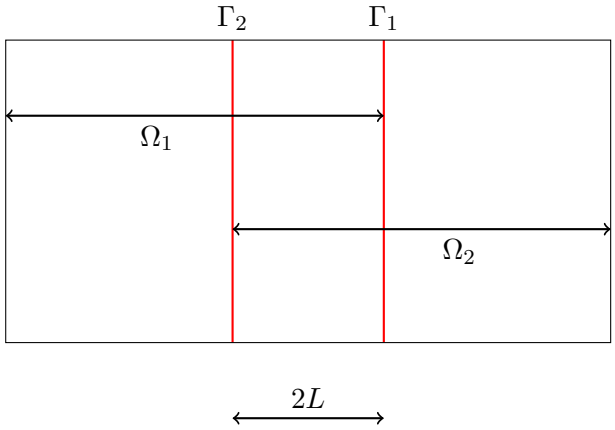

Figure 4-8: *Cartoon of the decomposition* $\Omega = (0, 1)^2$ *into two overlapping subdomains* Ω_1 *and* Ω_2.

of the Schwarz method we compute the residual by assembling the two subdomain approximations $\mathbf{u}_1^n$ and $\mathbf{u}_2^n$ to form a global approximation $\mathbf{U}^n$. In the overlapping region, or on the interface when $L = 0$, we take the average of the two subdomain approximations. We then compute the residual,

$$\mathbf{r}^n = \mathbf{1} - A_\epsilon \mathbf{U}^n.$$

The Schwarz iteration stops when the ℓ_2 norm of the residual reaches a given user defined tolerance, which we choose to be 10^{-6} in our numerical experiments (similar trends are observed for different tolerances). For all of the numerical experiments we fix ϵ and increase k with $h = \frac{\pi}{5k}$ and note the number of iterations taken by either the iterative Schwarz method or preconditioned GMRES.

In both sets of numerical tests we compare all of the choices of interface condition previously mentioned in Chapters 2, 3 and at the start of this chapter. We use the following notation to denote each choice

- S_0^T = zeroth order Taylor approximation (2.56).
- S_2^T = second order Taylor approximation (2.55).
- S_0^O = zeroth order optimised condition (3.2).
- S_2^O = second order optimised condition (4.8)

For the first set of experiments we set $L = 0$ and use a non-overlapping Schwarz method In Tables 4.1, 4.2, 4.3 and 4.4 we present the results using the Schwarz method as an iterative algorithm to solve the linear system (4.15). What we immediately see from these results is that the second order optimised condition performs the best out of all of the interface conditions. It is followed by the zeroth order optimised in terms of performance, then the second order Taylor approximation and finally the zeroth order Taylor approximation. The numerical results show that choosing a second order optimised interface condition results in convergence in at least a half of the number of iterations it takes for the zeroth order optimised and at most a third. We also notice that, as expected, when we increase ϵ the number of iterations needed to converge decreases for all of the methods, and once we reach $\epsilon = k^2$ each method converges independently (or nearly) of k, as k increases which is what is predicted by Corollary 3.4. A most interesting observation is that the second order Taylor approximation performs very well when $\epsilon = k^2$ almost converging in an many iterations as the zeroth order optimised method.

In figures 4-9 and 4-10 we plot the number of iterations from the previously mentioned tables against the corresponding k for the zeroth order and second order optimised methods. Along with this we plot the expected lower bounds on the number of iterations for k increasing found from Corollary 3.4 and the corresponding result using Conjecture 4.5. What we observe is that the numerical results agree with the behaviour of the theoretical bounds as k increases.

If we now examine Tables 4.5, 4.6, 4.7 and 4.8 these results show the number of GMRES iterations taken to solve the preconditioned system (4.16). What we observe is that we get an expected speed up in convergence in all of the methods as one would expect from using a preconditioned Krylov solver. In terms of the individual methods the second order optimised converges the fastest out of all four methods, again by almost half the iterations at best. We also observe the same expected behaviour when ϵ increases, namely that the number of iterations steadily decreases as ϵ increases. When $\epsilon = k^2$ we observe k independent convergence.

The next set of numerical experiments concern overlapping methods where we choose $L = h$. In Tables 4.9, 4.10, 4.11 and 4.12 we solve (4.15) using the Schwarz method as an iterative solver. We observe the same behaviour as we did with the non-overlapping

method, namely that the second order optimised interface condition results in a Schwarz method which converges in the least number of iterations. In comparison to both of the Taylor conditions it is a drastic improvement, and even converges at best in roughly a third of the number of iterations taken by the zeroth order optimised condition. Regarding the behaviour with respect to ϵ, we observe again that when ϵ increases the number of iterations decreases as does the growth in iterations as k increases. We also observe that when $\epsilon = k^2$ that we achieve convergence independent of k for all of the choices of interface condition. If we compare these results to those of the non-overlapping methods in tables 4.1, 4.2, 4.3 and 4.4 we see that even introducing the minimum amount of overlap, $L = h$ one grid cell, that this results in an improvement in convergence for all of the choices of methods. What can be seen is the improvement in convergence is more visible as k increases.

The results for solving the preconditioned system (4.16) using GMRES are given in Tables 4.13, 4.14, 4.15 and 4.16. We observe again the same speed up between these results and those of the iterative overlapping methods as we saw for the non-overlapping case. The most drastic improvement is seen for the both of the Taylor conditions. Whilst the convergence is improved for both of the optimised conditions it is not as striking. Indeed the second order optimised method works almost as well as an iterative method as it does when used as a preconditioner for GMRES. This is especially the case as ϵ approaches k^2 in magnitude. Finally if we compare the results of these tables to those in 4.5, 4.6, 4.7 and 4.8 we see that the addition of an overlap does improve convergence of preconditioned GMRES. The largest improvement is especially visible for low choices of ϵ and large k for the two Taylor methods.

Recalling from Section 2 that M_ϵ^{-1} is a good preconditioner for A when ϵ is chosen such that

(1) A_ϵ^{-1} is an effective preconditioner for A, so $\|\mathbb{I} - A_\epsilon^{-1}A\|_2$ is small,

(2) and that M_ϵ^{-1} is an effective preconditioner for A_ϵ, so $\|\mathbb{I} - M_\epsilon^{-1}A_\epsilon\|_2$ is small.

We can conclude that we have substantial numerical evidence to claim that (2) is satisfied when $\epsilon = k^2$. However from the results of [38] we know that (1) is satisfied when $\frac{\epsilon}{k}$ is sufficiently small. In the numerical experiments in the next section we examine numerically how to choose ϵ such that M_ϵ^{-1} is a good preconditioner for A.

These methods could of course be used for a decomposition of Ω into N_{sub} subdomains, where we would expect to see a growth in the number of iterations of both the iterative method and preconditioned GMRES as N_{sub} increased. We consider many subdomain decompositions in the next section and also in Chapter 5.

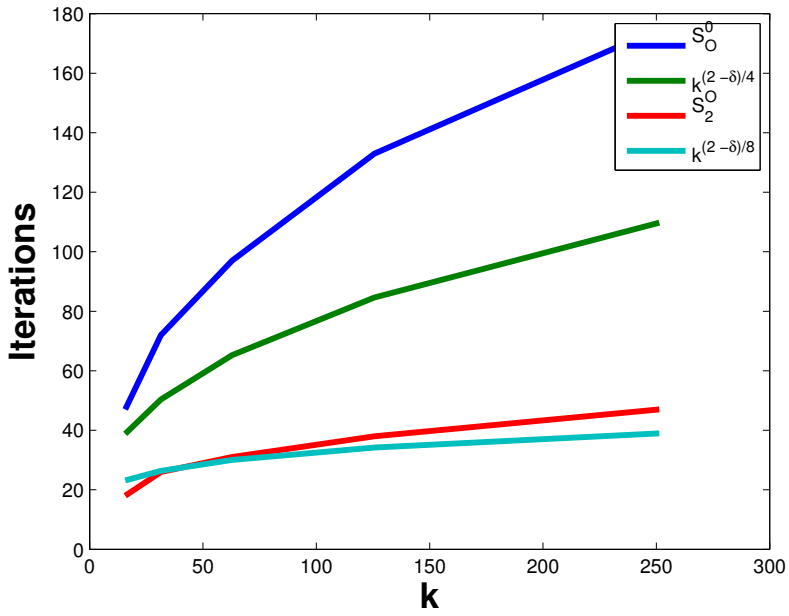

Figure 4-9: *A plot of the number of Schwarz iterations for increasing k for the method with optimised zeroth order (blue) and optimised second order (red) for $\epsilon = k^{\frac{1}{2}}$. These are compared with the theoretical bounds in dashed lines.*

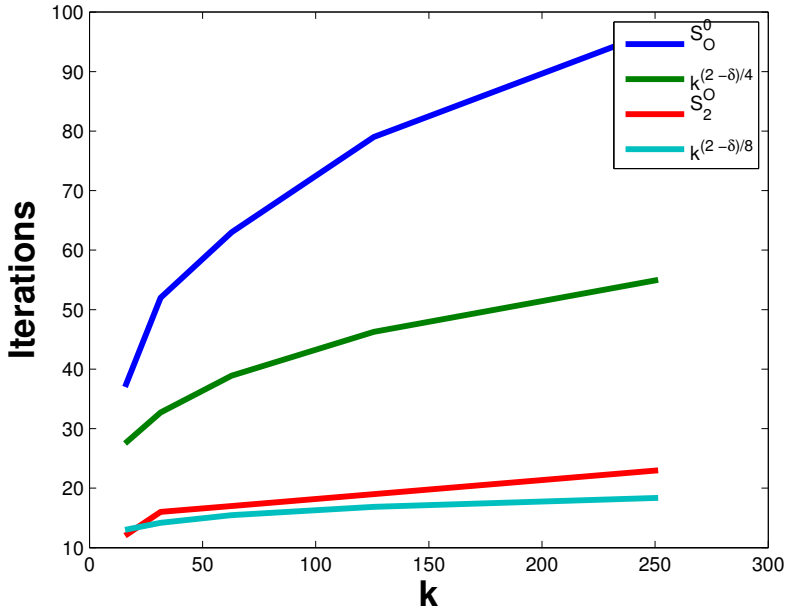

Figure 4-10: *A plot of the number of Schwarz iterations for increasing k for the method with optimised zeroth order (blue) and optimised second order (red) for $\epsilon = k$. These are compared with the theoretical bounds in dashed lines.*

k	size(A_ϵ)	Interface condition			
		S_0^T	S_2^T	S_0^O	S_2^O
5π	625	>1000	629	47	18
10π	2500	>1000	>1000	72	26
20π	10000	>1000	>1000	97	31
40π	40000	>1000	>1000	133	38
80π	160000	>1000	>1000	175	47

Table 4.1: *Number of Schwarz iterations for a fixed k, $h = \frac{\pi}{5k}$ and $\epsilon = k^{\frac{1}{2}}$.*

k	size(A_ϵ)	Interface condition			
		S_0^T	S_2^T	S_0^O	S_2^O
5π	625	>1000	208	37	13
10π	2500	>1000	546	52	16
20π	10000	>1000	>1000	63	17
40π	40000	>1000	>1000	79	19
80π	160000	>1000	>1000	97	23

Table 4.2: *Number of Schwarz iterations for a fixed k, $h = \frac{\pi}{5k}$ and $\epsilon = k$.*

k	size(A_ϵ)	Interface condition			
		S_0^T	S_2^T	S_0^O	S_2^O
5π	625	250	62	28	11
10π	2500	730	105	32	13
20π	10000	>1000	151	38	14
40π	40000	>1000	220	43	15
80π	160000	>1000	315	50	17

Table 4.3: *Number of Schwarz iterations for a fixed k, $h = \frac{\pi}{5k}$ and $\epsilon = k^{\frac{3}{2}}$.*

k	size(A_ϵ)	Interface condition			
		S_0^T	S_2^T	S_0^O	S_2^O
5π	625	78	21	19	11
10π	2500	130	23	22	11
20π	10000	138	24	22	12
40π	40000	138	24	23	12
80π	160000	146	25	23	12

Table 4.4: *Number of Schwarz iterations for a fixed k, $h = \frac{\pi}{5k}$ and $\epsilon = k^2$.*

k	size(A_ϵ)	Interface condition			
		S_0^T	S_2^T	S_0^O	S_2^O
5π	625	38	24	24	15
10π	2500	62	36	30	18
20π	10000	86	49	35	20
40π	40000	98	51	41	22
80π	160000	171	100	48	23

Table 4.5: *Number of GMRES iterations for a fixed k, $h = \frac{\pi}{5k}$ and $\epsilon = k^{\frac{1}{2}}$.*

k	size(A_ϵ)	Interface condition			
		S_0^T	S_2^T	S_0^O	S_2^O
5π	625	36	24	20	12
10π	2500	62	36	23	14
20π	10000	83	46	26	16
40π	40000	88	54	29	18
80π	160000	136	66	32	19

Table 4.6: *Number of GMRES iterations for a fixed k, $h = \frac{\pi}{5k}$ and $\epsilon = k$.*

k	size(A_ϵ)	Interface condition			
		S_0^T	S_2^T	S_0^O	S_2^O
5π	625	34	22	16	10
10π	2500	52	27	18	11
20π	10000	60	30	19	11
40π	40000	66	33	20	12
80π	160000	72	37	21	12

Table 4.7: *Number of GMRES iterations for a fixed k, $h = \frac{\pi}{5k}$ and $\epsilon = k^{\frac{3}{2}}$.*

k	size(A_ϵ)	Interface condition			
		S_0^T	S_2^T	S_0^O	S_2^O
5π	625	28	18	14	10
10π	2500	30	18	14	10
20π	10000	30	18	14	10
40π	40000	30	18	14	10
80π	160000	30	18	14	10

Table 4.8: *Number of GMRES iterations for a fixed k, $h = \frac{\pi}{5k}$ and $\epsilon = k^2$.*

k	size(A_ϵ)	Interface condition			
		S_0^T	S_2^T	S_0^O	S_2^O
5π	625	>1000	659	45	14
10π	2500	>1000	>1000	68	17
20π	10000	>1000	>1000	79	19
40π	40000	>1000	>1000	102	22
80π	160000	>1000	>1000	119	25

Table 4.9: *Number of Schwarz iterations for a fixed k, $h = \frac{\pi}{5k}$, $L = h$ and $\epsilon = k^{\frac{1}{2}}$.*

k	size(A_ϵ)	Interface condition			
		S_0^T	S_2^T	S_0^O	S_2^O
5π	625	898	204	36	12
10π	2500	>1000	535	49	14
20π	10000	>1000	>1000	58	15
40π	40000	>1000	>1000	68	17
80π	160000	>1000	>1000	81	19

Table 4.10: *Number of Schwarz iterations for a fixed k, $h = \frac{\pi}{5k}$, $L = h$ and $\epsilon = k$.*

k	size(A_ϵ)	Interface condition			
		S_0^T	S_2^T	S_0^O	S_2^O
5π	625	220	63	27	11
10π	2500	403	98	29	11
20π	10000	553	146	32	12
40π	40000	818	213	38	13
80π	160000	>1000	304	44	15

Table 4.11: *Number of Schwarz iterations for a fixed k, $h = \frac{\pi}{5k}$, $L = h$ and $\epsilon = k^{\frac{3}{2}}$.*

k	size(A_ϵ)	Interface condition			
		S_0^T	S_2^T	S_0^O	S_2^O
5π	625	78	21	18	10
10π	2500	78	23	18	10
20π	10000	79	23	19	10
40π	40000	82	24	20	10
80π	160000	84	24	20	10

Table 4.12: *Number of Schwarz iterations for a fixed k, $h = \frac{\pi}{5k}$, $L = h$ and $\epsilon = k^2$.*

k	size(A_ϵ)	Interface condition			
		S_0^T	S_2^T	S_0^O	S_2^O
5π	625	31	24	23	14
10π	2500	61	34	28	18
20π	10000	81	44	31	20
40π	40000	93	50	35	22
80π	160000	131	52	35	22

Table 4.13: *Number of GMRES iterations for a fixed k, $h = \frac{\pi}{5k}$, $L = h$ and $\epsilon = k^{\frac{1}{2}}$.*

k	size(A_ϵ)	Interface condition			
		S_0^T	S_2^T	S_0^O	S_2^O
5π	625	34	20	19	12
10π	2500	57	32	22	14
20π	10000	78	39	25	16
40π	40000	80	40	28	18
80π	160000	86	44	31	19

Table 4.14: *Number of GMRES iterations for a fixed k, $h = \frac{\pi}{5k}$, $L = h$ and $\epsilon = k$.*

k	size(A_ϵ)	Interface condition			
		S_0^T	S_2^T	S_0^O	S_2^O
5π	625	30	18	16	10
10π	2500	44	22	17	11
20π	10000	49	24	18	11
40π	40000	53	26	19	12
80π	160000	57	28	20	12

Table 4.15: *Number of GMRES iterations for a fixed k, $h = \frac{\pi}{5k}$, $L = h$ and $\epsilon = k^{\frac{3}{2}}$.*

k	size(A_ϵ)	Interface condition			
		S_0^T	S_2^T	S_0^O	S_2^O
5π	625	23	14	13	10
10π	2500	24	14	13	10
20π	10000	24	14	13	10
40π	40000	25	14	13	10
80π	160000	25	14	13	10

Table 4.16: *Number of GMRES iterations for a fixed k, $h = \frac{\pi}{5k}$, $L = h$ and $\epsilon = k^2$.*

4.3 Numerical experiments using the Schwarz method on multiple subdomains to solve $M_\epsilon^{-1}A = M_\epsilon^{-1}\mathbf{1}$

In our final set of experiments we examine the convergence of the Schwarz method with multiple subdomains used as a preconditioner for GMRES. We also compute the field of values of the resulting preconditioned linear system. We consider the following Helmholtz problem as our model problem

$$\left.\begin{aligned} -\Delta u - k^2 u = 1, \text{ in } \Omega = (0,1)^2 \\ \frac{\partial}{\partial n} u + iku = 0, \text{ on } \partial\Omega \end{aligned}\right\} \tag{4.17}$$

If we then discretise (4.17) with piecewise linear finite elements on a uniform grid with grid spacing h we obtain the following linear system,

$$A\mathbf{U} = \mathbf{1}. \tag{4.18}$$

We now precondition the above equation with M_ϵ^{-1} and solve the following preconditioned linear system with GMRES,

$$M_\epsilon^{-1}A\mathbf{U} = M_\epsilon^{-1}\mathbf{1}, \tag{4.19}$$

where M_ϵ^{-1} is the approximate inverse of A_ϵ (the matrix arising form the discretisation of (4.14)) formed using one iteration of a domain decomposition method. The domain decomposition method that we choose for our experiments is the Restricted Additive Schwarz method (RAS) [9], where our domain Ω is decomposed into N_{sub} overlapping subdomains. We choose to decompose Ω into strips, see Figure 4-11. The preconditioner that we construct using RAS is the following,

$$M_\epsilon^{-1} = \sum_{i=1}^{N_{sub}} \tilde{R}_i^T \left(\hat{A}_\epsilon\right)_i^{-1} R_i \tag{4.20}$$

where R_i is a restriction matrix which takes data on the whole of Ω and restricts it to just those nodes on the overlapping subdomain Ω_i. Similarly $\tilde{R}_i$ is a restriction matrix but for a non overlapping subdomain $\tilde{\Omega}_i$. Finally the matrices $(\hat{A}_\epsilon)_i$ are formed by discretising (4.14) (or this equation with optimised boundary conditions on the interface between subdomains, we choose the optimised zeroth order condition in our computations) on a local subdomain Ω_i. We then solve the preconditioned system (4.19) using GMRES-RAS with a tolerance of 10^{-7}.

In Tables 4.17, 4.18, 4.19 we look at the effect of varying the value of ϵ, the interface condition between subdomains, and the number of subdomain N_{sub} Throughout we

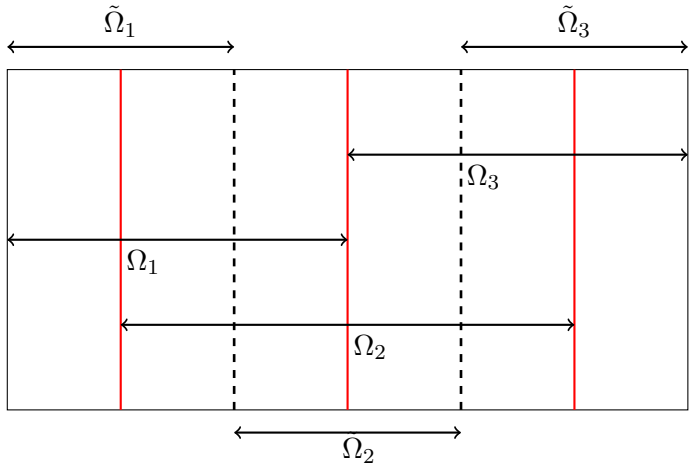

Figure 4-11: *Cartoon of the decomposition $\Omega = (0,1)^2$ into three overlapping subdomains Ω_1, Ω_2 and Ω_3. The non-overlapping domains are denoted by $\tilde{\Omega}_i$ for $i = 1, 2, 3$. The red lines represent the interfaces of the overlapping domains and the dashed lines the interfaces of the non-overlapping domains.*

solve (4.19) using GMRES-RAS for a fixed k, $hk < 1$ and an overlap between subdomains of h. When ϵ is increased we find that the convergence steadily deteriorates, and when $\epsilon = k^2$ the convergence is very poor. The optimal value of ϵ for a good GMRES-RAS solver seems to be $\epsilon \leq k$, though it should be noted that choosing $\epsilon = k^{\frac{1}{2}}$ results in only a few more iterations. The influence of the choice of interface condition was not as significant as was found in the previous section. Indeed the optimised interface condition (denoted by S_0^O) converged in at best ten to twenty iterations less than the standard impedance condition (denoted by S_0^T), but more often than not only in one or two iterations less. Finally as the number of subdomains was increased a deterioration in the convergence was observed, this however was to be expected as there was no coarse space correction [8, §2]. It would be expected that if a suitable coarse space was used then the number of iterations should significantly decrease as N_{sub} increases. However, the design of such coarse spaces was not a focus of this thesis, we refer the reader to the following reference for recent work on two level domain decomposition methods for the Helmholtz problem [34].

This convergence can be explained by examining the field of values of the matrix on the left hand side of (4.19). In Figures 4-12, 4-13, 4-14, 4-15 we examine how increasing ϵ influences the boundary of the field of values of $M_\epsilon^{-1}A$ for fixed k and $hk < 1$. In these computations M_ϵ^{-1} is formed by one iteration of RAS where the interface condition between subdomains is a zeroth order optimised condition (4.1). We find that as ϵ is increased this results in a decrease in the distance of the boundary of the field of values from the origin (as was observed previously in Section 2.2 for $A_\epsilon^{-1}A$). Therefore according to Theorem 2.3 we should expect GMRES to converge fastest when

$\epsilon = k$ (or indeed $\epsilon = k^{\frac{1}{2}}$) as the boundary of the field of values appears (at least from this numerical evidence) to be bounded away from the origin. This convergence of GMRES-RAS is observed in Tables 4.17, 4.18, 4.19. Hence we conclude that based on the numerical evidence that a choice of $\epsilon \sim k$ results in a domain decomposition preconditioner M_ϵ^{-1} which acts as a good preconditioner for A, see (2.15). In the large scale industrial and 3D computations in Chapter 5 we shall also use a choice of $\epsilon \sim k$ in our preconditioner, however it still remains an open question to prove this result theoretically.

N_{sub}	$\epsilon = k^{1/2}$		$\epsilon = k$		$\epsilon = k^{3/2}$		$\epsilon = k^2$	
	S_0^T	S_0^O	S_0^T	S_0^O	S_0^T	S_0^O	S_0^T	S_0^O
2	12	11	10	9	17	15	28	26
4	18	16	18	16	23	19	33	30
8	32	27	32	26	34	27	42	34
16	57	57	57	55	58	52	62	54

Table 4.17: *Number of GMRES iterations for a fixed number of subdomains N_{sub}. Here $k = 5\pi$, $h = \frac{\pi}{5k}$, $ovlp = h$.*

N_{sub}	$\epsilon = k^{1/2}$		$\epsilon = k$		$\epsilon = k^{3/2}$		$\epsilon = k^2$	
	S_0^T	S_0^O	S_0^T	S_0^O	S_0^T	S_0^O	S_0^T	S_0^O
2	15	13	13	11	23	21	60	57
4	24	21	23	21	32	29	68	63
8	40	42	40	42	46	45	80	74
16	88	79	88	75	90	71	109	90

Table 4.18: *Number of GMRES iterations for a fixed number of subdomains N_{sub}. Here $k = 10\pi$, $h = \frac{\pi}{5k}$, $ovlp = h$.*

N_{sub}	$\epsilon = k^{1/2}$		$\epsilon = k$		$\epsilon = k^{3/2}$		$\epsilon = k^2$	
	S_0^T	S_0^O	S_0^T	S_0^O	S_0^T	S_0^O	S_0^T	S_0^O
2	17	15	14	13	31	28	125	119
4	29	30	27	28	42	36	139	131
8	55	64	55	57	63	60	154	145
16	102	144	102	114	105	102	186	184

Table 4.19: *Number of GMRES iterations for a fixed number of subdomains N_{sub}. Here $k = 20\pi$, $h = \frac{\pi}{5k}$, $ovlp = h$.*

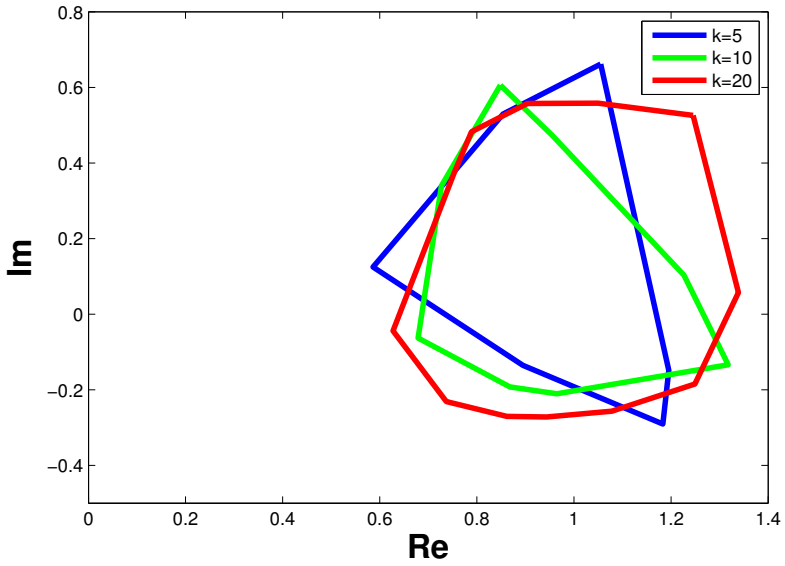

Figure 4-12: *The boundary of the field of values of* $M_\epsilon^{-1}A$ *for* $k = 5, 10, 20$ *with* $\epsilon = k^{\frac{1}{2}}$.

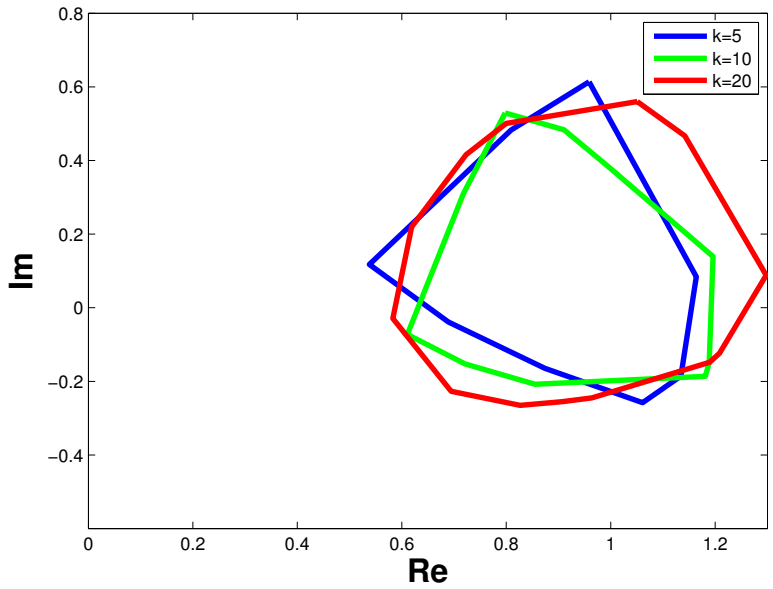

Figure 4-13: *The boundary of the field of values of* $M_\epsilon^{-1}A$ *for* $k = 5, 10, 20$ *with* $\epsilon = k$.

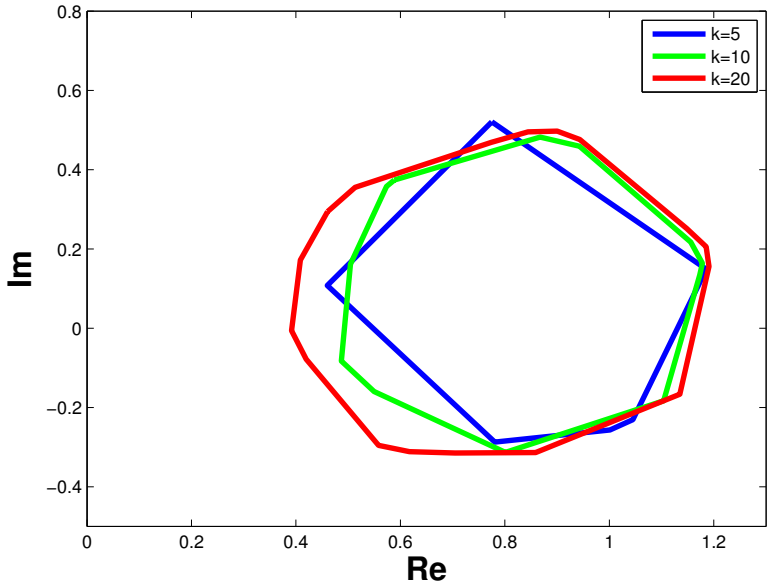

Figure 4-14: *The boundary of the field of values of* $M_\epsilon^{-1}A$ *for* $k = 5, 10, 20$ *with* $\epsilon = k^{\frac{3}{2}}$.

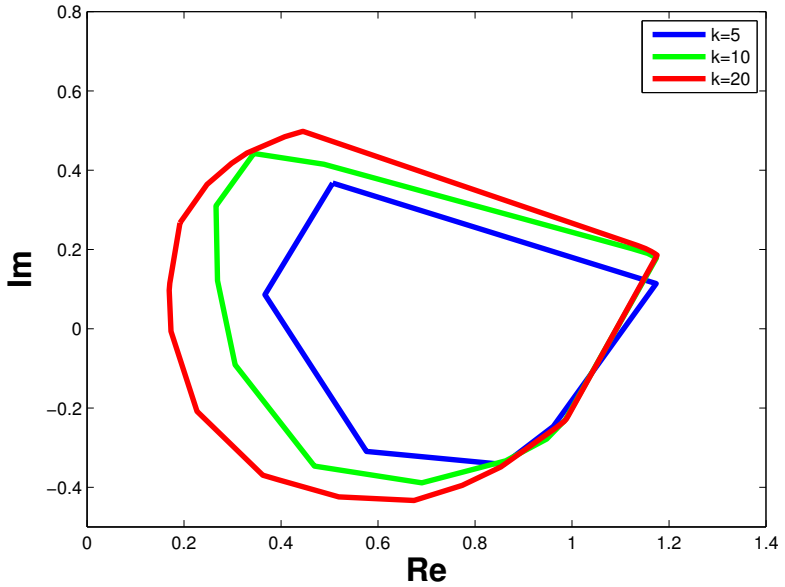

Figure 4-15: *The boundary of the field of values of* $M_\epsilon^{-1}A$ *for* $k = 5, 10, 20$ *with* $\epsilon = k^2$.

CHAPTER 5

THE SWEEPING PRECONDITIONER AND A NEW HYBRID DOMAIN DECOMPOSITION BASED VARIANT

In this chapter we develop a hybrid preconditioning method which is suitable for the iterative solution of large scale Helmholtz problems including 3D problems. This hybrid method uses a combination of the *sweeping* preconditioner (introduced in [5]) with an overlapping domain decomposition method. The domain decomposition methods that we use are those developed in Chapters 2, 3 and guided by the numerical experiments in Chapter 4. The *sweeping* preconditioner is an effective preconditioner for the iterative solution of the Helmholtz equation for constant and non constant wavenumber. However it does require the solution of substantial subproblems which can be costly. We shall use domain decomposition to approximate these subproblems The sweeping preconditioner was developed by Björn Engquist and Lexing Ying with two different approaches given in [5], [6]. It is the former approach, using moving perfectly matched layers, which we will discuss here.

Assuming that we have discretised our PDE with finite differences, on a rectangular domain and grid, then the algorithm works by constructing an approximate block LDL^t factorisation of the system matrix, where the matrix D is a block diagonal matrix whose diagonal elements are dense matrices. The block size is the corresponding number of unknowns in each row of the grid, assuming a lexicographical ordering. In this Chapter we choose to use finite differences as this was the choice of discretisation used by Dr Paul Childs in the software he developed at Schlumberger Gould research, therefore we used this convention though one could repeat these experiments using finite elements and obtain similar results. This factorisation of the system matrix could be inverted which results in a similar matrix factorisation of the form $(L^t)^{-1}S(L)^{-1}$, where S is a

block diagonal matrix whose elements are dense matrices. The essence of the sweeping preconditioner is an efficient approximation of this inverse (discussed in detail later). Effectively this method is then an approximation of the classical Thomas algorithm [53], written in block form. The idea of the algorithm presented [5] is to then approximate the elements of S to reduce the cost of the inversion. This will be discussed in detail later.

We first introduce the Thomas algorithm for solving symmetric tridiagonal linear systems which we shall relate to the sweeping approach of the authors [5]. The model problem and the concept of *perfectly matched layers (PML)* are then introduced. With these preliminaries we then introduce the sweeping factorisation and then finally present the main idea behind the preconditioner. For simplicity, we present the preconditioner in 2D but is valid also in 3D where it is also an effective preconditioner. Experiments in 3D are given later.

5.1 The Thomas algorithm for symmetric tridiagonal matrices

The Thomas algorithm [53], also known as the tridiagonal matrix algorithm, is a classical method for solving a linear system where the system matrix is tridiagonal and diagonally dominant (the algorithm is proved to be stable for this type of matrix). For clarity let us consider a linear system of the form $\tilde{A}\tilde{\mathbf{u}} = \mathbf{f}$ where,

$$\tilde{A} = \begin{pmatrix} a_{1,1} & a_{1,2} & 0 & 0 \\ a_{2,1} & a_{2,2} & \ddots & 0 \\ 0 & \ddots & \ddots & a_{n-1,n} \\ 0 & 0 & a_{n,n-1} & a_{n,n} \end{pmatrix}$$

where the system matrix $\tilde{A}$ is of size $n \times n$.

The Thomas algorithm essentially is based on the observation that $\tilde{A}$ has the factorisation

$$\tilde{A} = LDL^t. \tag{5.1}$$

In the above D is a diagonal matrix

$$D = \begin{pmatrix} s_{1,1} & 0 & \dots & 0 \\ 0 & s_{2,2} & \ddots & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & \dots & s_{n,n} \end{pmatrix} \tag{5.2}$$

whose entries are given by

$$\begin{aligned} s_{1,1} &= a_{1,1}, \\ s_{i,i} &= a_{i,i} - a_{i,i-1} s_{i-1,i-1}^{-1} a_{i-1,i}, \text{ for } i = 2, \dots, n \end{aligned}$$

and L is the lower triangular matrix

$$L = \begin{pmatrix} 1 & 0 & \dots & 0 \\ s_{1,1}^{-1} a_{2,1} & 1 & \ddots & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & s_{n-1,n-1}^{-1} a_{n,n-1} & 1 \end{pmatrix} \tag{5.3}$$

Based on this factorisation the algorithm for solving $\tilde{A}\tilde{\mathbf{u}} = \mathbf{f}$ proceeds by first solving the linear system $L\mathbf{v} = \mathbf{f}$ by forward substitution starting from the top row. We then perform a diagonal scaling by solving the system $D\mathbf{y} = \mathbf{v}$ and then finish by solving the linear system $L^t\tilde{\mathbf{u}} = \mathbf{y}$ by back substitution starting from the bottom row. This is shown in Algorithm 3. We can then count the elementary operations (el.ops) in Algorithm 3, where the construction of the LDL^t factorisation costs of $\mathcal{O}(n)$ [23]. Consider first the forward sweep in the Algorithm (where $s_{i-1,i-1}^{-1} a_{i-1,i}$ is a known value in L computed during the construction of the LDL^t factorisation). In line 4 we have 2 el.ops (one multiplication and subtraction) which we do $n-1$ times giving a total of $2(n-1)$ el.ops for the forward sweep. If we consider now the backward sweep we have 1 el.op in line 7 where we divide by $s_{n,n}$. Then in line 9 we have one subtraction, one multiplication and one division which are carried out $n-1$ times. Therefore there is a total of $3(n-1)+1$ in the backward sweep. Hence the total for application for all of the algorithm is $\mathcal{O}(n)$ el.ops.

Before introducing the analogue of this algorithm for a matrix which is symmetric block tridiagonal (the so called sweeping factorisation) we shall introduce our model problem and so called perfectly matched layers (PML) which can be a more accurate approximation to the Sommerfeld radiation condition compared to a simple impedance condition.

Algorithm 3 Application of Thomas algorithm to solve $\tilde{A}\mathbf{u} = \mathbf{f}$

%Forward sweep
$v_1 = f_1$
for $i = 2 \dots n$ **do**
    $v_i = f_i - \left(s_{i-1,i-1}^{-1} a_{i,i-1}\right) v_{i-1}$
end for
%Backward sweep combining diagonal scaling
$u_n = v_N s_{n,n}^{-1}$
for $i = n-1 \dots 1$ **do**
    $\tilde{u}_i = \left(v_i - a_{i,i+1} u_{i+1}\right) s_{i,i}^{-1}$
end for

5.2 Model problem

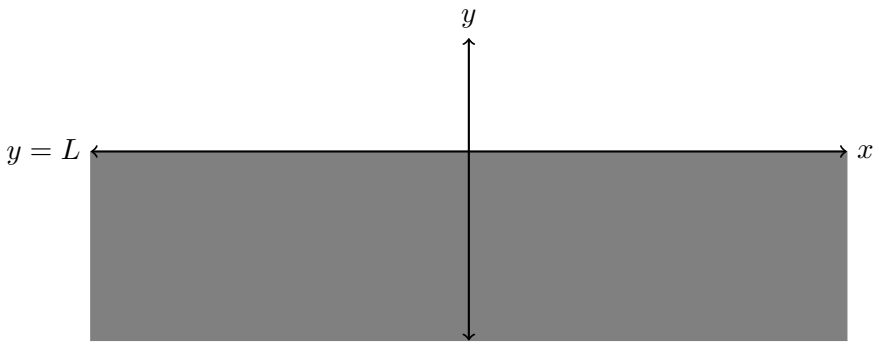

Figure 5-1: *Illustration of the half space considered, i.e. the region of* $\mathbb{R}^2$ *below* $y = L$.

We consider as our motivating problem the Helmholtz equation,

$$\mathcal{L}u(\mathbf{x}) := \left(-\Delta - k^2(\mathbf{x})\right) u(\mathbf{x}) = f(\mathbf{x}), \tag{5.4}$$

on the half space $\{(x, y) \in \mathbb{R}^2 : y < L\}$. We assume for simplicity a zero Dirichlet boundary condition at $y = L$, and to ensure that the problem is well posed the Sommerfeld radiation condition is imposed at infinity in both x and y. The wavenumber

$$k(\mathbf{x}) = \frac{\omega}{c(\mathbf{x})}, \tag{5.5}$$

is allowed to be variable, with ω denoting the angular frequency and $c(\mathbf{x})$ the wave speed.

In practice we approximate the above PDE by choosing a finite region of interest which we choose to be, without loss of generality, the square domain $\Omega = (0, L)^2$ for $L \in \mathbb{R}_+$. Our model problem is then to solve (5.4) on Ω with boundary conditions which approximate the Sommerfeld condition presented in (1.4). An example of this

is the impedance condition (1.5) but a better approximation can be obtained by the method of *perfectly matched layers*.

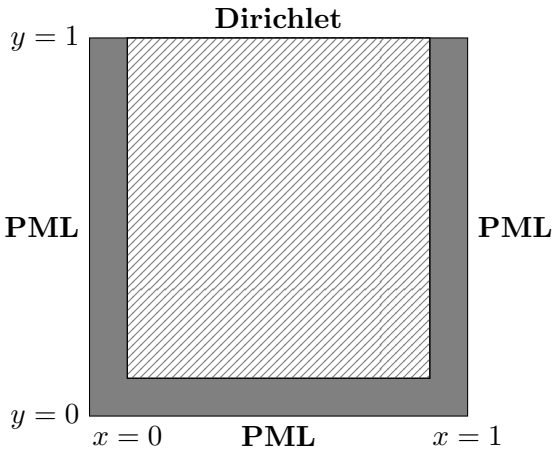

Figure 5-2: *Cartoon of model problem with PML region in grey.*

5.2.1 Perfectly matched layers

Perfectly matched layers (*PML*) were first introduced by Bérenger in [7]. Here we present the variant developed by Collino and Tsogka [20], which uses a complex coordinate transform. Normally for the Helmholtz equation one imposes an absorbing boundary condition, typically some approximation to the so called Sommerfeld radiation condition, which tries to ensure that waves radiate outwards (to infinity) from a source, and that there is no (or as little as possible) reflection at the boundary. In some sense the method of *PML* is not actually a boundary condition but rather it involves the introduction of an absorbing layer. A Dirichlet condition is placed on the outer boundary of the layer. The *PML* truncates the domain by introducing an artificial absorbing layer next to the boundary in which outgoing waves decay exponentially.

This method for the geometry depicted in Figure 5-2 can be constructed using the following transformation,

$$\frac{\partial}{\partial x} \text{ is replaced by } \theta_1(x)\frac{\partial}{\partial x}, \tag{5.6}$$

$$\frac{\partial}{\partial y} \text{ is replaced by } \theta_2(y)\frac{\partial}{\partial y}, \tag{5.7}$$

where,

$$\theta_1(x) = \frac{1}{1 + i\frac{\phi_1(x)}{\omega}},$$

$$\theta_2(y) = \frac{1}{1 + i\frac{\phi_2(y)}{\omega}},$$

and,

$$\phi_1(x) = \begin{cases} \frac{C_P}{\eta}\left(\frac{x-\eta}{\eta}\right)^2, & \text{if } 0 \le x \le \eta, \\ 0, & \text{if } \eta < x < 1-\eta, \\ \frac{C_P}{\eta}\left(\frac{x-1+\eta}{\eta}\right)^2, & \text{if } 1-\eta \le x \le 1, \end{cases}$$

$$\phi_2(y) = \begin{cases} \frac{C_P}{\eta}\left(\frac{y-\eta}{\eta}\right)^2, & \text{if } 0 \le y \le \eta, \\ 0, & \text{if } y > \eta. \end{cases}$$

Here η is our chosen PML width, typically of the order of a wavelength $\lambda = \frac{2\pi}{\omega}$, and $C_P > 0$ is to be chosen. The function ϕ_2 behaves similarly to ϕ_1 in Figure 5-3, but is zero in the far right region as there is a Dirichlet boundary present there in the discretisation. Physically what these θ functions do is that they introduce an artificial damping into the Helmholtz but only in the PML region (i.e. the region in grey in Figure 5-2). We can observe this from Figure 5-4. This artificial damping then forces the solution of the Helmholtz equation in the region PML region to decay exponentially as it reaches the outer boundary.

We can now use (5.6),(5.7) to rewrite our PDE (5.4) as one which includes a PML absorbing layer of width η.

5.2.2 The linear system and the sweeping factorisation

We now use our transformations (5.6), (5.7) to approximate our PDE (5.4) as the following,

$$\left(\theta_1(x)\frac{\partial}{\partial x}\left(\theta_1(x)\frac{\partial}{\partial x}\right) + \theta_2(y)\frac{\partial}{\partial y}\theta_2(y)\left(\frac{\partial}{\partial y}\right) + k^2(x,y)\right)u(x,y) = -f(x,y), \text{ in } \Omega,$$
$$u(x,y) = 0, \text{ on } \partial\Omega.$$

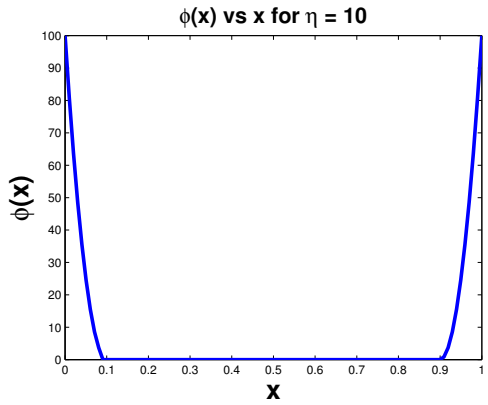

Figure 5-3: *Plot of* $\phi_1(x)$ *for* $x \in [0,1]$ *with* $C_p = 1$, *and* $\eta = 10$.

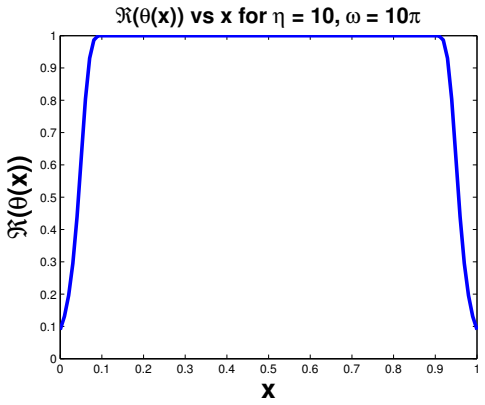

Figure 5-4: *Plot of* $\Re\left(\theta_1(x)\right)$ *for* $x \in [0,1]$ *with* ϕ *as in the left hand plot and* $\omega = 10\pi$.

We can rewrite the above PDE in a symmetric form by simply dividing through by $\theta_1(x)\theta_2(y)$ and noting that $\theta_1(x)$ is independent of y and $\theta_2(y)$ is independent of x to obtain,

$$\left.\begin{aligned}\left(\frac{\partial}{\partial x}\left(\frac{\theta_1}{\theta_2}\frac{\partial}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{\theta_2}{\theta_1}\frac{\partial}{\partial y}\right)+\frac{k^2}{\theta_1\theta_2}\right)u &= -\frac{f}{\theta_1\theta_2},\ \text{in } \Omega,\\ u &= 0,\ \text{on } \partial\Omega.\end{aligned}\right\} \tag{5.8}$$

where we have dropped the x, y dependence for brevity. The above PDE is the same as (5.4) when we are in the interior of Ω, away from the *PML* layer (i.e. the hatched region in Figure 5-2). But inside the *PML* we have a PDE with variable coefficients coming from the variation in the functions θ_1, θ_2. Moreover we have homogeneous Dirichlet conditions on the boundary $\partial\Omega$.

We choose to discretise (5.8) with a standard 5-point finite difference stencil on an equidistant grid with spacing $h = \frac{1}{n+1}$, where n is the number of grid points in the x or y direction, and hence the total number of grid points $N = n^2$.

In practice this simple discretisation scheme is not enough to achieve numerical accuracy. If hk is kept constant, which can often be the case in applications, then it is known that the accuracy of this scheme decreases as $k \to \infty$. This is known as the pollution effect [25]. In large scale applications (such as Seismic inversion) it is therefore common place for higher order numerical schemes to be used, such as those outlined in [27].

If we denote our grid by $G := \{(ih, jh),\ \text{where } 1 \leq i, j \leq n\}$, then we choose to order $\mathbf{u}_{i,j}$ and $\mathbf{f}_{i,j}$, the solution vector and the external force respectively, lexicographically row by row starting from the first row at $y = 0$. We therefore end up with a linear

system $A\mathbf{u} = \mathbf{f}$ to solve which has the following form,

$$\begin{pmatrix} A_{1,1} & A_{1,2} & 0 & 0 \\ A_{2,1} & A_{2,2} & \ddots & 0 \\ 0 & \ddots & \ddots & A_{n-1,n} \\ 0 & 0 & A_{n,n-1} & A_{n,n} \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_n \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_n \end{pmatrix} =: \mathbf{f} \tag{5.9}$$

where for $i = 1, \ldots, n$

$$\mathbf{u}_i = (u_{1,i}, u_{2,i}, \ldots u_{n,i})^t, \text{ and,}$$
$$\mathbf{f}_i = (f_{1,i}, f_{2,i}, \ldots f_{n,i})^t,$$

and the matrix A will be complex and symmetric i.e. $A_{i,i-1} = A_{i,i-1}^t$, but not Hermitian.

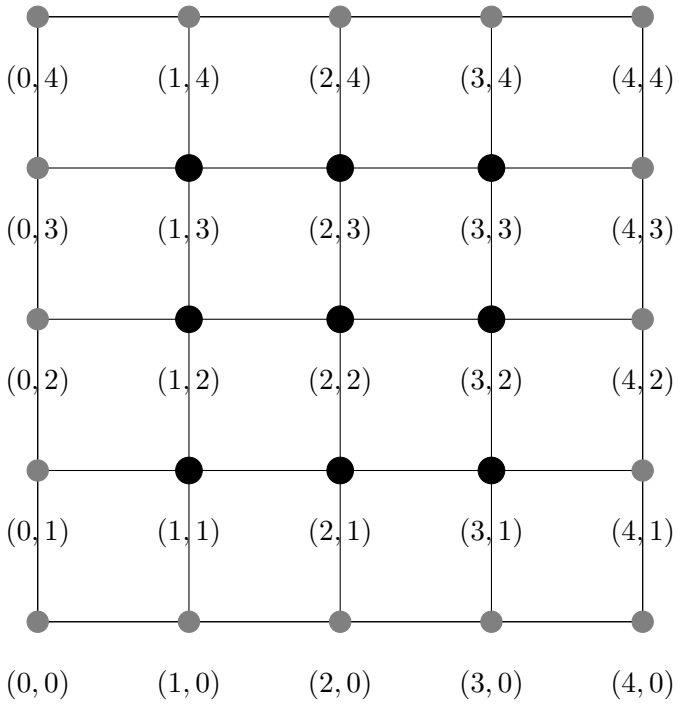

Figure 5-5: *Cartoon of lexicographical ordering. Here the unknowns are given by black nodes and the Dirichlet conditions given in grey.*

Let us show what the elements of A are by taking a simple example with the ordering given in Figure 5-5, where we have a Dirichlet condition on the outer grey nodes of Figure 5-5. If we disretise (5.8) using the standard 5 point finite difference then the

equation for $A\mathbf{u} = \mathbf{f}$ is given by the following at the node $(1,1)$,

$$\left(\frac{\theta_1}{\theta_2}\right)_{\frac{1}{2},1} u_{0,1} + \left(\frac{\theta_1}{\theta_2}\right)_{\frac{3}{2},1} u_{2,1} + \left(\frac{\theta_2}{\theta_1}\right)_{1,\frac{1}{2}} u_{1,0} + \left(\frac{\theta_2}{\theta_1}\right)_{1,\frac{3}{2}} u_{1,2} +$$
$$+ h^2 \left[\left(\frac{k^2}{\theta_1\theta_2}\right)_{1,1} - \left(\left(\frac{\theta_1}{\theta_2}\right)_{\frac{1}{2},1} + \left(\frac{\theta_1}{\theta_2}\right)_{\frac{3}{2},1} + \left(\frac{\theta_2}{\theta_1}\right)_{1,\frac{1}{2}} + \left(\frac{\theta_2}{\theta_1}\right)_{1,\frac{3}{2}}\right)\right] u_{1,1} = \frac{h^2 f_{1,1}}{\theta_1\theta_2}.$$

Because of the zero Dirichlet conditions on the boundary we have $u_{0,1} = u_{1,0} = 0$. Hence at node $(1,1)$ we have the equation,

$$\left(\frac{\theta_1}{\theta_2}\right)_{\frac{3}{2},1} u_{2,1} + \left(\frac{\theta_2}{\theta_1}\right)_{1,\frac{3}{2}} u_{1,2} +$$
$$+ h^2 \left[\left(\frac{k^2}{\theta_1\theta_2}\right)_{1,1} - \left(\left(\frac{\theta_1}{\theta_2}\right)_{\frac{1}{2},1} + \left(\frac{\theta_1}{\theta_2}\right)_{\frac{3}{2},1} + \left(\frac{\theta_2}{\theta_1}\right)_{1,\frac{1}{2}} + \left(\frac{\theta_2}{\theta_1}\right)_{1,\frac{3}{2}}\right)\right] u_{1,1} = \frac{h^2 f_{1,1}}{\theta_1\theta_2}.$$

Therefore the matrix A will have the following entries to start with,

$$\begin{pmatrix} h^2\left[\left(\frac{k^2}{\theta_1\theta_2}\right)_{1,1} - \left(\left(\frac{\theta_1}{\theta_2}\right)_{\frac{1}{2},1} + \left(\frac{\theta_1}{\theta_2}\right)_{\frac{3}{2},1} + \left(\frac{\theta_2}{\theta_1}\right)_{1,\frac{1}{2}} + \left(\frac{\theta_2}{\theta_1}\right)_{1,\frac{3}{2}}\right)\right] & \left(\frac{\theta_1}{\theta_2}\right)_{\frac{3}{2},1} & 0 & \dots \\ \left(\frac{\theta_1}{\theta_2}\right)_{\frac{3}{2},1} & \star & \star & \dots \\ 0 & \ddots & \ddots & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{pmatrix}.$$

Here we can see that given the example grid in Figure 5-5 A is a block 3×3 matrix and each block is of size 3×3. In this simple example each plane of the grid has 3 points (there are 5 but 2 are by the Dirichlet conditions) which then form the blocks of the matrix A, as we see in the above equation. We can then fill out the rest of A in a similar fashion.

We note here that the diagonal blocks $A_{i,i}$ are tridiagonal and the upper and lower diagonal blocks $A_{i,i-1}$, $A_{i,i+1}$ are diagonal.

One way to approach the direct solution of (5.9) is to construct a block LDL^t factorisation. We consider this using a row by row elimination starting from $y = 0$ (see Figure 5-6). This process starts on the first row at $y = 0$,

$$\begin{pmatrix} A_{1,1} & A_{1,2} & 0 & 0 \\ A_{2,1} & A_{2,2} & \ddots & 0 \\ 0 & \ddots & \ddots & A_{n-1,n} \\ 0 & 0 & A_{n,n-1} & A_{n,n} \end{pmatrix} = L_1 \begin{pmatrix} S_1 & 0 & 0 & 0 & 0 \\ 0 & S_2 & A_{2,3} & 0 & 0 \\ 0 & A_{3,2} & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & \ddots & A_{n-1,n} \\ 0 & 0 & 0 & A_{n,n-1} & A_{n,n} \end{pmatrix} L_1^t$$

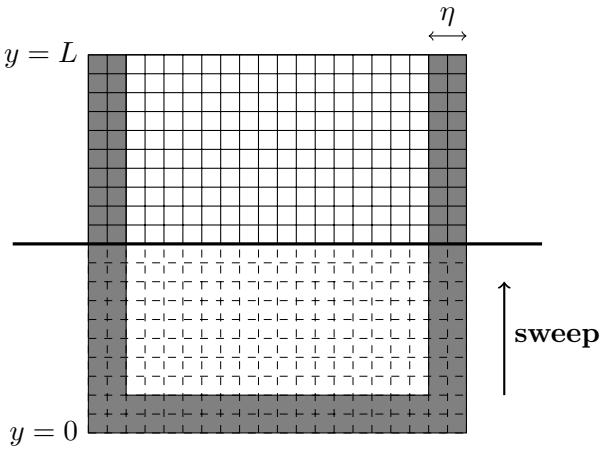

Figure 5-6: *Cartoon of sweeping action, with PML width η.*

where $S_1 = A_{1,1}$, $S_2 = A_{2,2} - A_{2,1}S_1^{-1}A_{1,2}$ and,

$$L_1 = \begin{pmatrix} \mathbb{I} & 0 & 0 & 0 \\ A_{2,1}S_1^{-1} & \mathbb{I} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \mathbb{I} \end{pmatrix}, \quad L_{n-1} = \begin{pmatrix} \mathbb{I} & 0 & 0 & 0 \\ 0 & \mathbb{I} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & A_{n,n-1}S_{n-1}^{-1} & \mathbb{I} \end{pmatrix}, \tag{5.10}$$

and where $\mathbb{I}$ denotes the $n \times n$ identity matrix. This process is repeated for all rows to give the following factorisation,

$$A = LDL^t. \tag{5.11}$$

In the above equation D is given by

$$D = \begin{pmatrix} S_1 & 0 & 0 & 0 \\ 0 & S_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & S_n \end{pmatrix} \tag{5.12}$$

where S is a Schur complement of the form,

$$\begin{aligned} &S_1 = A_{1,1}, \\ &S_i = A_{i,i} - A_{i,i-1}S_{i-1}^{-1}A_{i-1,i}, \text{ for } 2 \leq i \leq n, \end{aligned} \tag{5.13}$$

and where,

$$L = (L_1, \ldots, L_{n-1}) = \begin{pmatrix} \mathbb{I} & 0 & 0 & 0 \\ A_{2,1}S_1^{-1} & \mathbb{I} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & A_{n,n-1}S_{n-1}^{-1} & \mathbb{I} \end{pmatrix}, \tag{5.14}$$

Now if we want to find the solution of our linear system then we can invert this factorisation to find the solution, via,

$$\mathbf{u} = \left(L_1^t\right)^{-1} \ldots \left(L_{n-1}^t\right)^{-1} \begin{pmatrix} S_1^{-1} & 0 & 0 & 0 \\ 0 & S_2^{-1} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & S_n^{-1} \end{pmatrix} L_{n-1}^{-1} \ldots L_1^{-1}\mathbf{f}. \tag{5.15}$$

On comparison with the matrices (5.2),(5.3) defined previously for the Thomas algorithm it is apparent that (5.12) and (5.14) are their block equivalents. Therefore the above sweeping factorisation (5.11) is just a block tridiagonal version of the Thomas algorithm of Section 5.1, where the matrix in the left hand side of (5.9) must be block diagonally dominant. During the construction of the factorisation (5.11) the dominant cost will be incurred from constructing the S_i matrices. We can see this is true by recalling the definition of S_i in (5.13). In this definition we see that constructing S_i involves inverting of the matrix $S_{i-1,i-1}$ which is very costly. These S_i matrices therefore will be dense and of size $n \times n$. Therefore the cost of constructing a matrix S_i will be of order $\mathcal{O}(n^3)$, if we perform the inversion of S_{i-1}^{-1} exactly. We do this n times during the factorisation hence the cost of constructing this factorisation (5.11) is $\mathcal{O}(n^4) = \mathcal{O}(N^2)$. More important is the inversion of the factorisation, given by (5.15), which can be used to compute the solution of our linear system. This inversion process costs of order $\mathcal{O}(n^3) = \mathcal{O}(N^{\frac{3}{2}})$, assuming that S_i^{-1} are known. To show this let us write (5.15) more concisely as $L^{-t}S^{-1}L^{-1}$. The matrix L^{-t} is involves multiplication by the dense matrices $\left(L_i^t\right)^{-1}$ which costs $\mathcal{O}(n^2)$, and this has to be done n times hence $\mathcal{O}(n^3)$. Similarly for L^{-1}. Between L^{-t} and L^{-1} we multiply by the dense matrices S_i^{-1} which costs $\mathcal{O}(n^2)$ and again we do this n times. Hence the overall cost of the factorisation is $\mathcal{O}(n^3) = \mathcal{O}(N^{\frac{3}{2}})$. Therefore this method is in itself not very useful for computing $\mathbf{u}$, as this is as expensive as solving $A\mathbf{u} = \mathbf{f}$ with a direct solver in 2D which is in general $\mathcal{O}(N^{\frac{3}{2}})$ [26]. The main cost in construction and application is due to the Schur complements S_i, we will in the next section discuss a method to approximate the S_i and reduce the computational cost.

Algorithm 4 Algorithm for the application of (5.15)

%Forward sweep
for $i = 1 \dots n-1$ **do**
  $u_{i+1} = f_{i+1} - A_{i+1,i}\left(S_i^{-1} u_i\right)$
end for
%Backward sweep
for $i = n-1 \dots 1$ **do**
  $u_i = S_i^{-1} u_i - S_i^{-1}\left(A_{i,i+1} u_{i+1}\right)$
end for

5.3 The moving PML method

As the computation of these Schur complement S_i matrices and their inverses are the bottleneck in the implementation of the sweeping preconditioner this leads one to ask if there is some reasonable way to approximate them to reduce the cost.

The approach that Engquist and Ying took, to find a suitable approximation, was to examine how these inverse Schur complement matrices relate to the Green's function of the Helmholtz equation in the half space in Figure 5-1. This is a heuristic approach based on physical intuition, however it results in an effective approximation to these inverse Schur complement matrices.

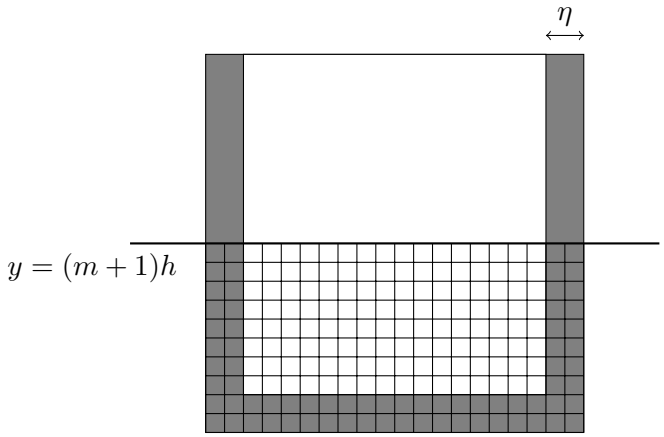

Figure 5-7: *Cartoon of restriction to upper $m \times n$ block.*

If we now restrict ourselves to only the top left $m \times n$ block of A, then the problem that we are solving is on region $\Omega_m = [0,1] \times [0,(m+1)h]$ shown in Figure 5-7. Computing

the LDL^t factorisation as in (5.11) gives,

$$\begin{pmatrix} A_{1,1} & A_{1,2} & 0 & 0 \\ A_{2,1} & A_{2,2} & \ddots & 0 \\ 0 & \ddots & \ddots & A_{m-1,m} \\ 0 & 0 & A_{m,m-1} & A_{m,m} \end{pmatrix} = L_1 \dots L_{m-1} \begin{pmatrix} S_1 & 0 & 0 & 0 \\ 0 & S_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & S_m \end{pmatrix} L_{m-1}^t \dots L_1^t. \tag{5.16}$$

What we can observe from (5.16) is that the matrix on the left hand side is an approximate disretisation of the Helmholtz equation (5.4) on the half plane with Dirichlet boundary at $y = (m+1)h$ and with the infinite region truncated on three sides by a *PML*. If we then invert (5.16) this gives,

$$\begin{aligned} &\begin{pmatrix} A_{1,1} & A_{1,2} & 0 & 0 \\ A_{2,1} & A_{2,2} & \ddots & 0 \\ 0 & \ddots & \ddots & A_{m-1,m} \\ 0 & 0 & A_{m,m-1} & A_{m,m} \end{pmatrix}^{-1} \\ &= \left(L_1^t\right)^{-1} \dots \left(L_{m-1}^t\right)^{-1} \begin{pmatrix} S_1^{-1} & 0 & 0 & 0 \\ 0 & S_2^{-1} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & S_m^{-1} \end{pmatrix} L_{m-1}^{-1} \dots L_1^{-1}. \end{aligned} \tag{5.17}$$

It is from (5.17) that Engquist and Ying make their central observations which motivate the two approaches that they take in [5], [6] the former of which we discuss here. These observations are:

(1). The LHS of (5.17) can be thought of as a discrete half space Green's function.

Explanation:
We first recall that the left hand side matrix of (5.16) is the discrete form of the Helmholtz equation with zero boundary condition at $y = (m+1)h$ and PML condition on all the other boundaries, see Figure 5-7, which we will denote $A^{(m)}$. Therefore if we invert this matrix and multiply this with the source vector $\mathbf{f}^{(m)}$ we get the solution on this $m \times m$ region of Ω,

$$\mathbf{u}^{(m)} = \left(A^{(m)}\right)^{-1} \mathbf{f}^{(m)}. \tag{5.18}$$

Recall that we have defined our Helmholtz equation as $\mathcal{L}u = f$, where $\mathcal{L}$ is our Helmholtz operator, from (5.4). Then we can write the solution u as the convolu-

tion of the Green's function G and the source f,

$$u = G \star f. \tag{5.19}$$

Therefore we can see by comparing the discrete (5.18) and the continuous (5.19) problems that our matrix $\left(A^{(m)}\right)^{-1}$ is a discrete approximation of the Green's function of the Helmholtz problem on the half space with Dirichlet boundary at $y = (m+1)h$.

(2).The $(m,m)^{th}$ block of the RHS of (5.17) is equal to S_m^{-1}.

Explanation:
We can observe the above statement by looking at the form the L matrices take in (5.14). We recall from (5.10) that the matrices L_i, for $1 \leq i \leq m-1$, have identity matrices on their diagonal blocks and $A_{i+1}S_i^{-1}$ at the $(i+1,i)^{th}$ block. If we define $L = L_1 \dots L_{n-1}$ then the right hand side of (5.17) is,

$$\left(L^t\right)^{-1} \begin{pmatrix} S_1^{-1} & 0 & 0 & 0 \\ 0 & S_2^{-1} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & S_m^{-1} \end{pmatrix} L^{-1}.$$

Moreover,

$$L = \begin{pmatrix} \mathbb{I} & 0 & 0 & 0 \\ A_2S_1^{-1} & \mathbb{I} & 0 & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & A_mS_{m-1}^{-1} & \mathbb{I}. \end{pmatrix}$$

and,

$$L^{-1} = \begin{pmatrix} \mathbb{I} & 0 & 0 & 0 \\ -A_2S_1^{-1} & \mathbb{I} & 0 & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & -A_mS_{m-1}^{-1} & \mathbb{I}. \end{pmatrix}$$

Therefore if we write the right hand side of (5.17), with the L matrices written in block form, gives

$$\begin{pmatrix} \mathbb{I} & -A^2S_1^{-1} & 0 & 0 \\ 0 & \mathbb{I} & \ddots & 0 \\ 0 & 0 & \ddots & -A_mS_{m-1}^{-1} \\ 0 & 0 & 0 & \mathbb{I} \end{pmatrix} \begin{pmatrix} S_1^{-1} & 0 & 0 & 0 \\ 0 & S_2^{-1} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & S_m^{-1} \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 & 0 & 0 \\ -A_2S_1^{-1} & \mathbb{I} & 0 & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & -A_mS_{m-1}^{-1} & \mathbb{I} \end{pmatrix}$$

If the above is multiplied out then one can see that it is indeed true that the $(m,m)^{th}$ block is equal to S_m^{-1}

The consequence of these two observations is that the the matrix S_m^{-1} provides an approximation to the half space Green's function, and then evaluated at $y = mh$, with Dirichlet boundary condition at $y = (m+1)h$. However, the application of S_m^{-1} which is defined using (5.13) necessarily involves computations on the previous $(m-1)$ layers, and hence is very costly.

Figure 5-8: *Cartoon of reduced problem using artificial moving PML.*

One way of approximating S_m^{-1}, using the ideas mentioned previously, is the so called *moving PML* method. The idea of the *moving PML* method is to approximate S_m^{-1} by moving the PML at $y = 0$ up until it is just a few layers below $y = mh$. Why should we do this? If we recall the idea of the PML method, we use it to approximate an unbounded problem with a bounded one with $PMLs$ around the area of the unbounded problem in which we are interested. Therefore when we apply S_m^{-1}, our area of interest is only the layer at $y = mh$. Then instead of looking at the problem on the whole of the region $\Omega_m = [0,1] \times [0,(m+1)h]$. We now look at one on the smaller region $\Omega_b = [0,1] \times [(m-b)h,(m+1)h]$ close to $y = mh$, where b is a small number of layers to be chosen. The advantage of this is that the resulting approximation of S_m^{-1} in the sweeping inversion (5.15) involves solving a $b \times n$ PML problem where b is a constant number of layers. Therefore we are in effect solving a quasi-$1D$ problem on the region Ω_b which greatly reduces the computational cost of the sweeping inversion. We now approximate all the S_i^{-1} needed in implementing (5.13), by solving a series of $b \times n$ PML problems. The practical choice of b is a constant which is independent of n. In practice it is often chosen to be the same as the PML width, or of the order of a wavelength.

We now discuss the cost of applying the corresponding approximation of the sweeping factorisation with moving PML problem. We denote by $A^{(b)}$ the $bn \times bn$ Helmholtz matrix corresponding to discretisation on Ω_b. Now recalling that what we are interested in approximating the S_m^{-1} matrices by solving a linear system with the matrix $A^{(b)}$. The nodes in Ω_b are then reordered in the y direction where $P^{(b)}$ is the permutation matrix which induces this reordering. After this our discrete Helmholtz problem on Ω_b, has the system matrix $P^{(b)}A^{(b)}(P^{(b)})^t$ on Ω_b, with the new ordering of nodes. We then note that $P^{(b)}A^{(b)}(P^{(m)})^t$ is a banded matrix with bandwidth of $\mathcal{O}(b)$. The matrix $P^{(b)}A^{(b)}(P^{(b)})^t$ can then factorised using an LU factorisation i.e. $P^{(b)}A^{(m)}\left(P^{(b)}\right)^t =$

$L^{(b)}U^{(b)}$. Therefore the application involves the solution of a system of the form,

$$L^{(b)}U^{(b)}v^{(b)} = g^{(b)}, \tag{5.20}$$

where $v^{(b)}$ is a vector of the solution data on Ω_b and $g^{(b)}$ a vector of source data only on Ω_b. We can see that as b is chosen to be much smaller than n then we are solving a quasi-1D problem. As $L^{(b)}U^{(b)} = P^{(b)}A^{(b)}\left(P^{(m)}\right)^t$ is banded with $\mathcal{O}(b)$, solving the above linear system (5.20) exactly by Gaussian elimination will cost $\mathcal{O}(b^2n)$ [56]. Hence the overall cost will be $\mathcal{O}(b^2n^2) = \mathcal{O}(b^2N)$ as we perform these solves n times. Then as b is a constant, which one fixes, the cost of the inversion using the moving PML method can be thought of as linear with respect to the total number of grid points N. Of course this is not an exact inversion of the original problem but an approximation which can be used as a preconditioner for the original problem, or a related system.

In applications the preconditioner will be used with an iterative solver, such as GMRES. Therefore as the cost of applying the action of the *sweeping* preconditioner is $\mathcal{O}(b^2N)$, then the cost of the GMRES solver will be of $\mathcal{O}(N_{iter}b^2N)$, where N_{iter} is the number of iterations required to reach a specified tolerance. Hence if the N_{iter} is independent of N then preconditioned GMRES will have linear complexity. There is no theoretical evidence to prove this, but we shall provide numerical evidence later.

5.3.1 Preconditioning with the moving PML method

We shall illustrate the use of this sweeping inversion with *moving PML* as a preconditioner for solving the Helmholtz problem (5.4), with the same boundary conditions as Figure 5-2. Then, when discretised, as previously mentioned, this will result in a $n^2 \times n^2$ linear system $Au = f$. We use as a preconditioner to this system the approximate solution of the related system,

$$\mathcal{L}_\epsilon u(\mathbf{x}) := \left(-\Delta - k^2(\mathbf{x}) - i\epsilon\right) u(\mathbf{x}) = f(\mathbf{x}), \tag{5.21}$$

where $\epsilon > 0$ and the boundary conditions are the same as with the original problem. When discretised with finite differences (5.21) results in a $n \times n$ linear system $A_\epsilon \mathrm{u} = \mathbf{f}$, We then denote by P_ϵ the action of the approximation of A_ϵ^{-1} by the inversion process (5.15) using *moving PMLs* to approximatie the matrices S_i^{-1}. Then we solve the following preconditioned linear system,

$$P_\epsilon A\mathbf{u} = P_\epsilon \mathbf{f}, \tag{5.22}$$

using GMRES [57] with a set relative residual as its exit criterion.

In the next section we investigate how the complexity of this algorithm is affected by the choice of ϵ.

5.4 Numerical experiments with the sweeping preconditioner

In the previous section we have introduced the sweeping preconditioner [5], and shown that the complexity of the construction and application of the sweeping method was improved by introducing artificial moving PMLs in the interior of the computational domain. We now present novel numerical experiments where various parameters are varied to test the robustness of the preconditioned iterative solver. It is hoped that this will give some insight into why the authors of [5] chose certain parameter values and the robustness of this preconditioner.

The MATLAB code used to perform the numerical experiments was kindly provided by Dr Lexing Ying.

5.4.1 Experiments with the value of ϵ

We start by looking at how varying the parameter ϵ influences the robustness of the preconditioner.

Throughout these numerical studies we are solving the preconditioned linear system (5.22) iteratively using GMRES (without restarts) with an exit criterion that the relative residual reaches 10^{-6}. We also recall that the matrix A and vector $\mathbf{f}$ are formed by approximating the PDE (5.8) using the 5-point finite difference formula, and P_ϵ represents the action of applying the inversion process (5.15) with *moving PMLs* to A_ϵ with particular choices of ϵ. Throughout the experiments we choose a PML width of 12 grid points. Therefore the PML width is given by $\eta = 12h$ (where η is first mentioned in Section 5.2.1) with $h = \frac{1}{n+1}$, and $n = N^2$ where N is the total number of grid points.

Recall that the precondtioner uses the approximate solution to problem (5.21) with a PML boundary. In [5] the authors use something different but roughly equivalent,

$$\tilde{\mathcal{L}}u(\mathbf{x}) := \left(-\Delta - \tilde{k}^2\right) u(\mathbf{x}) = f(\mathbf{x}), \tag{5.23}$$

where $\tilde{k} := \frac{\omega + i\alpha}{c(\mathbf{x})}$ with ω as the angular frequency, $c(\mathbf{x})$ the wave speed. and α a

parameter to be chosen. We can see the similarity by looking at the simplest case by setting $c(\mathbf{x}) = 1$. If we then expand (5.23) gives,

$$\tilde{k}^2 := k^2 + 2i\alpha k - \alpha^2$$

Then comparing this to $k^2 + i\epsilon$, we can see that if the parameter α is chosen of $\mathcal{O}(1)$, as is it is chosen in [5], then this is equivalent to a choice of $\epsilon = \mathcal{O}(k)$ in (5.21). In the following experiments we shall explain the effect of varying ϵ.

Experiments on the unit square

We start with some experiments on the unit square, $\Omega = (0,1)^2$, with constant and non-constant wave number.

In the following experiments we choose an external force $f(x,y)$ which is a Gaussian point source,

$$f(x,y) = \exp\left(-\left(\frac{\omega}{\pi}\right)^2\left[(x-x_1)^2 + (y-y_1)^2\right]\right) \tag{5.24}$$

at the point $(x_1, y_1) = (\frac{1}{8}, \frac{1}{2})$ and we recall that $k := \frac{\omega}{c}$ where ω is the angular frequency and c is the wave speed. This choice of source generates circular waves propagating outwards from this point, as can be seen in Figure 5-10. In the experiments we increase the value of $\frac{\omega}{2\pi}$ and record the number of iterations taken for preconditioned GMRES to achieve a relative residual of 10^{-6}. The computational time taken to set up the factorisation (5.11) and the computational time taken by the solver are also recorded. Throughout the number of points per wavelength is fixed at 8 and hence the total number of grid points $N = (8\omega)^2$. This may seem low but is the choice of the authors in [5]. As mentioned previously we chose a PML width $\eta = 12h$. The width of the artificial PMLs in the sweeping preconditioner was chosen as $b = 2\eta$. For each example we choose $\epsilon = 0$, k, and k^2 and present the results in a separate table.

We start by choosing $c(\mathbf{x}) = 1$, the results of which can be seen in Tables 5.1, 5.2 and 5.3 and the numerical solution in Figure 5-9. The first observation we can make is that a choice of $\epsilon = 0$ results in a number of GMRES iterations which is relatively low and shows a slow increase as $\frac{\omega}{2\pi}$ is increased. When $\epsilon = k$ the total number of iterations is slightly larger, by at most 3 iterations, but as $\frac{\omega}{2\pi}$ is increased the number of iterations stays constant. Finally if $\epsilon = k^2$ then we observe a linear growth in the number of iterations as $\frac{\omega}{2\pi}$ increases.

We can also observe from the results of Tables 5.1, 5.2 and 5.3 that the computational times for the setup of the factorisation and the time to solve are consistent with

what was predicted earlier. The solve time increases linearly as N increases which is consistent with the $\mathcal{O}(N)$ complexity quoted earlier.

These experiments are repeated for a problem where the wave number, $k(\mathbf{x}) := \frac{\omega}{c(\mathbf{x})}$, varies throughout the domain $\Omega = (0,1)^2$. The variation in $c(\mathbf{x})$ is shown in Figure 5-10 (this was based on a model used in [5]) and the corresponding numerical solution for $\frac{\omega}{2\pi} = 64$ is show in Figure 5-11. The results are presented in Tables 5.4, 5.5 and 5.6 and show the same behaviour observed by Engquist and Ying, namely that a choice of $\epsilon = k$ results in a number of iterations which remains constant as $\frac{\omega}{2\pi}$ is increased. The poor performance of the method when $\epsilon = k^2$, in Table 5.6, being more marked than when $c(\mathbf{x})$ was constant.

What is rather surprising from these numerical tests is that a choice of $\epsilon = 0$ gives a performance which is as good if not better than that with $\epsilon = k$. However the benefit of introducing non zero ϵ in the preconditioner will be made more apparent from the numerical results using the more complicated Marmousi model.

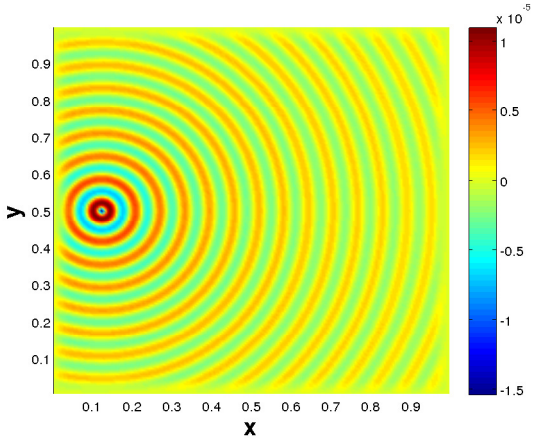

Figure 5-9: *Plot of* $u(x)$ *with* $\frac{\omega}{2\pi} = 16$ *for* $c(x) = 1$ *on* $(0,1)^2$.

$\frac{\omega}{2\pi}$	$N = n^2$	Iterations	Setup time	Solve time
8	64^2	3	6.615100e-02	6.400800e-02
16	128^2	4	2.100480e-01	1.470300e-01
32	256^2	5	9.639050e-01	8.463310e-01
64	512^2	7	4.046340e+00	4.746688e+00

Table 5.1: *Number of iterations using Sweeping preconditioner with moving PML as a preconditioner for GMRES with a tolerance of* 10^{-6}. *Here* $c(x) = 1$ *and* $\epsilon = 0$.

$\frac{\omega}{2\pi}$	$N = n^2$	Iterations	Setup time	Solve time
8	64^2	6	5.399400e-02	3.148200e-02
16	128^2	6	2.109030e-01	1.362780e-01
32	256^2	6	9.535790e-01	8.114470e-01
64	512^2	6	3.875587e+00	4.808404e+00

Table 5.2: *Number of iterations using Sweeping preconditioner with moving PML as a preconditioner for GMRES with a tolerance of* 10^{-6}. *Here* $c(x) = 1$ *and* $\epsilon = k$.

$\frac{\omega}{2\pi}$	$N = n^2$	Iterations	Setup time	Solve time
8	64^2	15	5.431800e-02	2.645240e-01
16	128^2	23	2.147850e-01	8.425990e-01
32	256^2	38	9.630820e-01	5.921758e+00
64	512^2	68	3.938831e+00	5.567415e+01

Table 5.3: *Number of iterations using Sweeping preconditioner with moving PML as a preconditioner for GMRES with a tolerance of* 10^{-6}. *Here* $c(x) = 1$ *and* $\epsilon = k^2$.

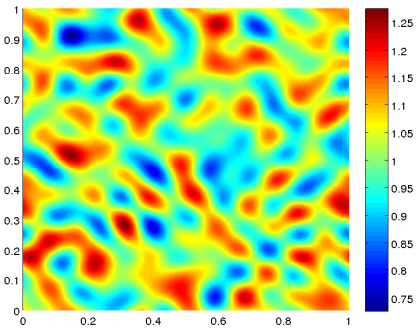

Figure 5-10: *Plot of $c(x)$ which is highly variable. This model comes from [5].*

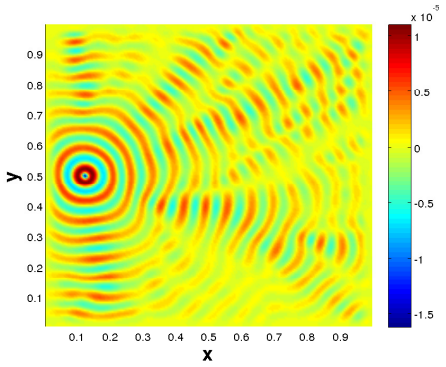

Figure 5-11: *Plot of $u(x)$ with $\frac{\omega}{2\pi} = 16$ for $c(x)$ given on $(0,1)^2$.*

$\frac{\omega}{2\pi}$	$N = n^2$	Iterations	Setup time	Solve time
8	64^2	8	6.700000e-02	1.231940e-01
16	128^2	7	2.143230e-01	2.609420e-01
32	256^2	8	1.043329e+00	1.260795e+00
64	512^2	10	3.802815e+00	5.690341e+00

Table 5.4: *Number of iterations using Sweeping preconditioner with moving PML as a preconditioner for GMRES with a tolerance of 10^{-6}. Here $c(x)$ is as given in Figure 5-10 and $\epsilon = 0$.*

$\frac{\omega}{2\pi}$	$N = n^2$	Iterations	Setup time	Solve time
8	64^2	9	5.336700e-02	1.136010e-01
16	128^2	10	2.144610e-01	3.380940e-01
32	256^2	10	9.582660e-01	1.257109e+00
64	512^2	11	3.787285e+00	6.821324e+00

Table 5.5: *Number of iterations using Sweeping preconditioner with moving PML as a preconditioner for GMRES with a tolerance of 10^{-6}. Here $c(x)$ is as given in Figure 5-10 and $\epsilon = k$.*

$\frac{\omega}{2\pi}$	$N = n^2$	Iterations	Setup time	Solve time
8	64^2	24	5.418200e-02	4.018250e-01
16	128^2	41	2.222340e-01	2.127200e-01
32	256^2	93	9.574110e-01	1.557937e+01
64	512^2	116	3.781013e+00	1.033074e+02

Table 5.6: *Number of iterations using Sweeping preconditioner with moving PML as a preconditioner for GMRES with a tolerance of 10^{-6}. Here $c(x)$ is as given in Figure 5-10 and $\epsilon = k^2$.*

Experiments with the Marmousi model

We now test the sweeping preconditioner on a more challenging model namely the Marmousi model [54]. This data set, created by Institut Français du Pétrol, is used as a velocity model (in our case wave velocity is $c(\mathbf{x})$) whose geometry and data are based on a region of the Cuanza river in Angola. We have chosen, for ease of computations, to take a 3000km^2 section of the original model which still contains all relevant wave speeds from 1.5km/s up to 5.5km/s. This section of the original model is shown in Figure 5-12 and the corresponding numerical solution using the 5 point finite difference stencil, for $\frac{\omega}{2\pi} = 15$ is shown in Figure 5-13 and a point source at the location $(x, y) = (1500, 500)$.

We repeat the same experiments as the previous subsection but using this more complicated velocity model given in Figure 5-10, we fix the PML width $\eta = 12h$ where h is the mesh spacing. The width of the artificial PMLs, b, in the sweeping preconditioner was chosen as $b = 2\eta$. The velocity model used has $N = 250^2$ total grid points. We increase the value of $\frac{\omega}{2\pi}$ from 5, 10, 15 with a fixed value of ϵ and record the total number of iterations taken by preconditioned GMRES to reach a relative residual of 10^{-6} and the time taken to solve, and also the time taken to set up the factorisation. The results are given in Tables 5.7, 5.8, 5.9, 5.10.

What we observe is similar to the previous experiments in the previous subsection. A choice of $\epsilon = k$ results in the fewest number of iterations for each choice of $\frac{\omega}{2\pi}$ and the slowest growth as $\frac{\omega}{2\pi}$ is increased. In Table 5.9 the poor performance with a choice of $\epsilon = k^2$ is more striking with the number of iterations increasing rapidly as $\frac{\omega}{2\pi}$ increases.

In Table 5.10 we obtain results using the same absorption as was chosen in [5]. We include these results to test whether the authors convention was indeed the optimal one for this preconditioner. If we compare the results in Tables 5.7 and 5.10 we see that the authors' choice does indeed seem to be best choice as the number of iterations are fewer and show no increase from $\frac{\omega}{2\pi} = 10$ to $\frac{\omega}{2\pi} = 15$.

The conclusion that we can draw from these sets of numerical experiments is that a choice of $\epsilon = \mathcal{O}(k)$ seems to be the optimal choice of absorption for the performance of the preconditioner. This choice then leads to a number of iterations of preconditioned GMRES which is very close to being independent of the wavenumber as was observed in [5].

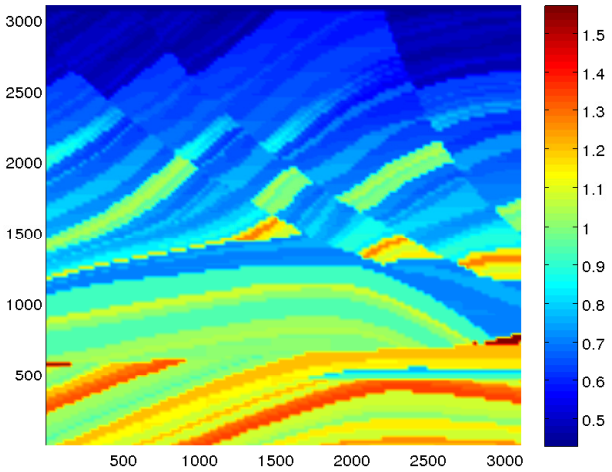

Figure 5-12: *Plot of $c(x)$ for the part of the Marmousi model used.*

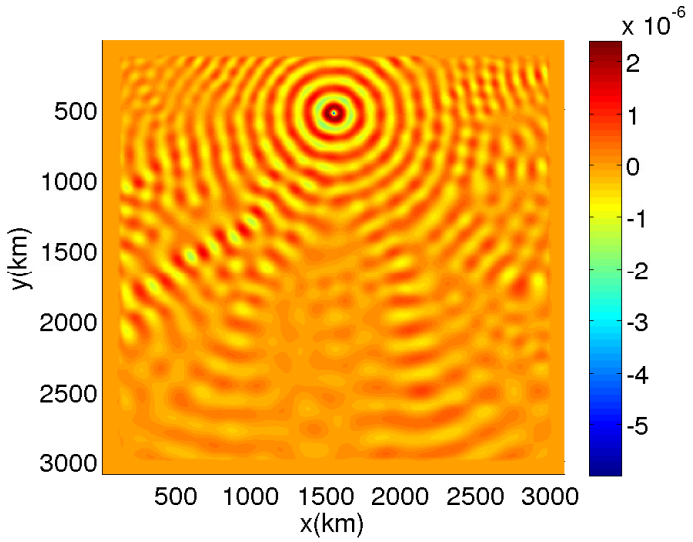

Figure 5-13: *Plot of $u(x)$ with $\frac{\omega}{2\pi} = 15$ for $c(x)$ given on the left.*

$\frac{\omega}{2\pi}$	$N = n^2$	Iterations	Setup time	Solve time
5	250^2	70	9.108250e-01	2.786235e+01
10	250^2	77	9.193370e-01	3.134366e+01
15	250^2	91	8.979550e-01	3.830419e+01

Table 5.7: *Number of iterations using Sweeping preconditioner with moving PML as a preconditioner for GMRES with a tolerance of* 10^{-6}*. Here* $c(x)$ *is the Marmousi model as given in Figure 5-12 and* $\epsilon = 0$*.*

$\frac{\omega}{2\pi}$	$N = n^2$	Iterations	Setup time	Solve time
5	250^2	63	9.144510e-01	2.688730e+01
10	250^2	71	9.559670e-01	3.036592e+01
15	250^2	77	9.216200e-01	3.451478e+01

Table 5.8: *Number of iterations using Sweeping preconditioner with moving PML as a preconditioner for GMRES with a tolerance of* 10^{-6}*. Here* $c(x)$ *is the Marmousi model as given in Figure 5-12 and* $\epsilon = k$*.*

$\frac{\omega}{2\pi}$	$N = n^2$	Iterations	Setup time	Solve time
5	250^2	130	9.281040e-01	3.914691e+01
10	250^2	214	9.024920e-01	1.239306e+02
15	250^2	313	8.885980e-01	2.442074e+02

Table 5.9: *Number of iterations using Sweeping preconditioner with moving PML as a preconditioner for GMRES with a tolerance of* 10^{-6}*. Here* $c(x)$ *is the Marmousi model as given in Figure 5-12 and* $\epsilon = k^2$*.*

$\frac{\omega}{2\pi}$	$N = n^2$	Iterations	Setup time	Solve time
5	250^2	60	8.903020e-01	2.293606e+01
10	250^2	69	9.005670e-01	2.501174e+01
15	250^2	69	9.363260e-01	2.650840e+01

Table 5.10: *Number of iterations using Sweeping preconditioner with moving PML as a preconditioner for GMRES with a tolerance of* 10^{-6}*. Here* $c(x)$ *is the Marmousi model as given in Figure 5-12 and* $k \to k + i\alpha$ *where* $\alpha = 1$*.*

5.5 Hybrid sweeping method

In this section we shall introduce a new hybrid preconditioner for the numerical solution of the Helmholtz equation, which combines the *sweeping* preconditioner of Engquist and Ying with a domain decomposition method. This method was mainly developed by Paul Childs of Schlumber Gould Research with the aid of Prof Ivan Graham and the author [46].

We shall first motivate this different approach, then introduce the algorithm itself, and then present numerical experiments testing the robustness of this new method.

5.5.1 Introduction

In the previous section we introduced the *sweeping* preconditioner in $2D$ and mentioned briefly that the ideas extended to $3D$. In industry applications we are concerned with the $3D$ numerical solution of the Helmholtz equation (5.4) and therefore we shall briefly discuss the sweeping preconditioner in $3D$ and quote its complexity.

As previously mentioned, the ideas introduced in the previous section in $2D$ follow quite intuitively to the problem in $3D$. For example if our grid is $(0, L)^d$, for $d = 2, 3$ then if one computes the sweeping factorisation in $2D$ (5.11) the process involves forming the blocks of the system matrix starting from the bottom of the $2D$ grid at $y = 0$ up to $y = L$, see Figure 5-6. In $3D$ we form the blocks of the system matrix in a similar fashion but now in planes of the grid $(0, L)^3$, where our moving PML regions are now $n \times n \times b$. We first recall that the application of the sweeping preconditioner to a problem on a $2D$

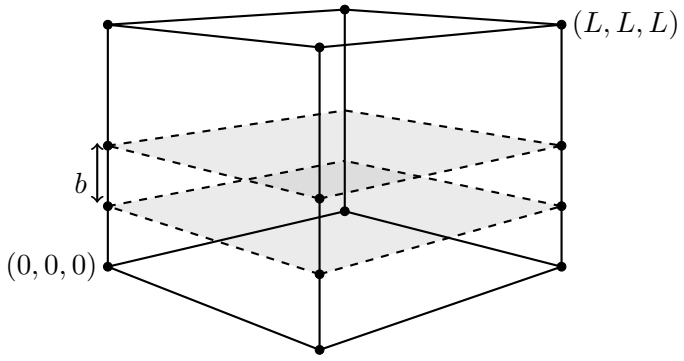

Figure 5-14: *Cartoon of the moving PML planes in $3D$. The moving PML now sweeps up the bottom face, the current moving PML region is between the two planes given in grey.*

grid, where the process involves solving quasi-$1D$ PML problems using a direct solver. If our grid is then in $3D$, a cube, the application of the preconditioner now involves the solution of quasi-$2D$ PML problems using a direct solver. Therefore the standard

sweeping preconditioner loses its linear complexity when we are solving a problem in $3D$ when using direct solvers to solve the quasi-$2D$ problems which arise from the moving PML layers. Engquist and Ying mention, [5], that linear complexity can be recovered by using the moving PML method, or the Hierarchical matrix framework [6], to solve the quasi-$2D$ problems arising in the application of the $3D$ *sweeping* preconditioner. If this is performed then the cost becomes $\mathcal{O}(b^3N)$. However the authors have not, to our knowledge, pursued this idea. Dr Paul Childs has implemented this method and found it to not to lend itself readily to parallelism.

Recently work has been done to devise ways to make the sweeping preconditioner scalable in parallel for very large $3D$ models. Dr Jack Poulson in his PhD thesis [47] and subsequent publications [32] introduced a new sparse direct solver to solve the linear systems from the moving PML regions in the sweeping preconditioner algorithm. This then allowed for a scalable parallel implementation of the sweeping preconditioner which was then applied successfully to solve a selection of large scale industrial problems.

In the following subsection we shall introduce a hybrid preconditioner involving the sweeping preconditioner and a domain decomposition method, the motivation for this approach being that due to the high memory costs of parallel direct solvers, preconditioned iterative methods seem like an attractive option.

We remind the reader that the motivation for this hybrid method, and indeed the overall goal of all Helmholtz solvers, is to efficiently solve $3D$ problems of size of the order 10^9, and work is ongoing at Schlumberger to improve this method.

5.5.2 The Hybrid method

For the standard sweeping preconditioner a direct solver is used to solve the linear systems arising from the moving PML regions in $3D$ we shall instead use an inner iterative method (such as GMRES or BiCGSTAB) preconditioned with a domain decomposition method.

Let us quickly restate our model problem, for clarity. We are concerned with numerical solutions of the Helmholtz equation (5.4) on a domain $\Omega \subset \mathbb{R}^3$, with Dirichlet boundary condition on the top face of the cube, and PML conditions on all other faces. We then discretise with finite difference (equally one could use finite elements) on an equidistant grid. The result is that we solve a linear system of the form,

$$A\mathbf{u} = \mathbf{f},$$

using GMRES. To improve convergence we precondition the above system, as mentioned in section 5.3.1, that is we solve the preconditioned linear system,

$$P_{\alpha_1} A\mathbf{u} = P_{\alpha_1}\mathbf{f},$$

where P_{α_1} denotes the action of approximating $A_{\alpha_1}^{-1}$ by the sweeping precondtioner with moving PMLs. Note here that we have an α_1 instead of α which we had in the notation previously, the reason for this is that there is another level of preconditioning to follow and hence another choice of α at this level. We note that A_α is the discretisation of the following PDE,

$$\mathcal{L}_\alpha u(\mathbf{x}) := \left(-\Delta - \frac{(\omega + i\alpha)^2}{c^2(\mathbf{x})}\right) u(\mathbf{x}) = f(\mathbf{x}).$$

using finite differences with PML boundary conditions.

The next step is to introduce a further level of preconditioning. The action of the preconditioner P_{α_1} requires the solution of linear systems of the form $A_{\alpha_1}\mathbf{u} = \mathbf{f}$ on domains of size $n \times n \times b$. Instead of using a direct solver to solve these linear systems, we use instead an iterative solver preconditioned with a domain decomposition method. Hence on the moving PML regions we solve, with an inner iterative method (with a truncated number of iterations), the following linear system,

$$M_{\alpha_2} A_{\alpha_1} \tilde{\mathbf{u}} = M_{\alpha_2} \tilde{\mathbf{f}}, \tag{5.25}$$

where M_{α_2} denotes the action of approximating $A_{\alpha_2}^{-1}$ with a domain decomposition method using ideas discussed in previous chapters. Note here that $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{f}}$ are the solution and the residual arising from the inner solves. We recall that this domain de-

composition approach amounts to firstly dividing the moving PML region into smaller subdomains see Figure 5-15. We then solve linear systems of the form $A_{\alpha_2}\tilde{\mathbf{u}} = \tilde{\mathbf{f}}$ restricted to these subdomains and then reassembling the solution on the whole of the moving PML region (5.25).

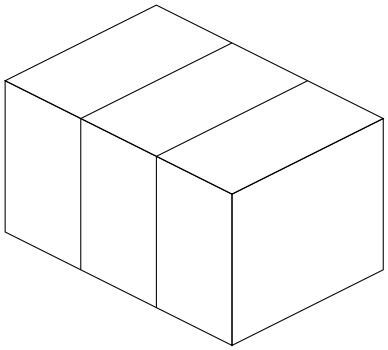

Figure 5-15: *Cartoon of simple* $3 \times 3 \times 1$ *domain decomposition of a* $3D$ *moving* PML *region.*

There are quite a few parameters present which have to be chosen in order to maximise the performance of the hybrid method: the best choice of α_1 and α_2, the choice of interface condition in the domain decomposition (with overlap) preconditioner, and the best combination of iterative methods for both levels of preconditioning. This will be addressed in the following subsections before we apply the hybrid method to some large scale industrial problems. It must be noted that the caveat is that there is no theory to this numerical method to date, and hence the conclusions that we reach are guided by numerical experiments.

An example of this is the choice of absorption parameter α for the inner (α_2) and outer (α_1) preconditioners. In [46] it was found through numerical experimentations that for a convergent algorithm a choice of $\alpha_2 \geq \alpha_1$ was required, and that a choice of

$$\alpha_1 \sim \frac{\alpha_2}{2},$$

was the most robust choice, and is our convention throughout the subsequent numerical experiments. Similar numerical investigations were performed to test the type of inner outer iterative methods used and the convergence criterion for the inner iterative method, details of which are present in [46], we shall present the final outcomes in a table in the following subsection.

5.5.3 Set up for Numerical experiments

We now list the choice of parameters used in the numerical experiments which will follow. As previously stated the optimal choice of each was found through numerical

experimentation and heuristics. The only parameter choice listed above not mentioned

Parameter	Value
α_1	0.5
α_2	1.0
PML width	$5h$
inner solver	GMRES
outer solver	FGMRES [50]
Max inner iterations	20
Max inner tolerance	10^{-4}
Moving PML width, b	$20h$
Number of planes swept at a time	7
DDM preconditioner	Restricted Additive Schwarz (RAS)
Overlap in Domain Decomposition	h

Table 5.11: *List of parameter choices used in later numerical tests, where h is the grid spacing.*

previously is the number of planes which are swept at a time. If we return to Algorithm 4 the index i denotes the current plane that we are sweeping through. We instead choose to sweep through several planes at once which can help to reduce the cost of the algorithm.

For the convergence criterion of the inner solves it was decided, through numerical experiments, that a sufficient criterion to end the inner iterative solver was when the Euclidean norm of the residual be less than or equal to 10^{-4}, or that we reach 20 GMRES iterations. How far we were able to truncate the inner solves was very important as to how effective the hybrid method performs. The outer tolerance was arbitrary and is noted in each set of numerical experiments.

The domain decomposition method of choice was restricted additive schwarz (RAS) [9], the reason for this is that out of the additive variants it was the most effective in terms of the convergence of the inner solver and overall cost. The reason an additive method was chosen over a multiplicative method was due to how easily additive methods lend themselves to parallelism, which is crucial for $3D$ problems.

In the domain decomposition method we used the minimal overlap possible of only one grid point. The reason for this was that in this was necessary in the larger computations to reduce the computational cost. If we were to include a larger overlap then one would expect to observe a reduction in the overall number of iterations but also an increase in computational time taken.

Finally it is worth mentioning that all numerical experiments were run at Schlumberger Gould Research on their 64 Bit Linux cluster using OpenMPI and Infiniband with Intel Xeon processors.

5.5.4 Choice of interface condition for DDM preconditioner

To start this subsection we remind the reader of the numerous notation for the interface conditions introduced in previous chapters in the following table. As we are performing

Interface condition	Description
S_0^T	Zeroth order Taylor series approximation
S_0^O	Zeroth order optimised by continuous analysis
S_0^{DO}	Zeroth order optimised by discrete analysis
S_2^T	Second order Taylor series approximation
S_2^O	Second order optimised by continuous analysis
S_2^{DO}	Second order optimised by discrete analysis

Table 5.12: *Type of interface condition and description.*

computations with velocity models which are highly heterogeneous the theory developed in Chapters 2, 3, for constant wavenumber (and hence constant velocity), can be used as guidance but we should not expect to observe the same the same trends in iteration counts. We do not include PML interface conditions, the reason for this is that in preliminary numerical experiments this choice performed poorly in comparison to the other choices presented above. However a method to optimise the PML boundary condition for a domain decomposition method is currently being investigated by Dr Vladimir Druskin and Dr Paul Childs [45]. Initial numerical results show that this approach could further improve convergence of the inner domain decomposition preconditioner.

The choice of interface condition which we arrived at was the first order discrete optimised condition S_0^{DO}. We present two sets of numerical results to justify this claim in Tables 5.13, 5.14, where we note the number of iterations taken by the outer solver and the overall CPU time as we increase the number of subdomains in the domain decomposition. Both sets of experiments use a Gaussian point source (5.24) in $2D$ this is located at $\mathbf{x} = (500, 1000)$ and in $3D$ at $\mathbf{x} = (6000, 6000, 50)$.

For the results of Table 5.13 we use the hybrid method to solve the $2D$ Helmholtz equation with the so called BP-EAGE velocity model [19] shown in Figure 5-16, the real part of the numerical solution is given in Figure 5-17. These computations were actually the most challenging of all of the computations due to the large jumps present in the velocity model, see Figure 5-16. The total number of grid points for this model (including PMLs) is $N = 814 \times 90 = 73260$. In Table 5.13 it is clear that S_0^{DO} is the best in terms of the number of iterations and CPU times. The only surprising results are that the second order conditions perform rather poorly in comparison to the zeroth order conditions. Though most surprising is that S_0^O fails to converge in 100 outer GMRES iterations, this seems to suggest that the inner iterative method has

not converged sufficiently and is not proving to be an effective preconditioner for the outer iterative method. A large increase in the number of iterations was observed as the number of subdomains increases. The reason for this was probably due to the fact that the subdomain sizes became too small and therefore the numerical solution on each subdomain was not accurate enough.

In the next table 5.14 we use the hybrid method to solve the Helmholtz equation with the $3D$ SEG-Salt velocity model [19] (shown in Figure 5-18 the real part of the numerical solution is shown in Figure 5-19). The total number of grid points (including PMLs) for this model is $N = 146 \times 146 \times 52 = 1108432$. An illustration of the domain decomposition in each swept domain is given in Figure 5-15. The $3D$ results in Table 5.14 agree with the $2D$ results, in that a choice of S_0^{DO} in the inner domain decomposition preconditioner results in a smaller number of outer GMRES iterations and CPU time. However, in these results the number of iterations and CPU time do not vary as greatly as in the $2D$ results which seems to suggest that the inner iterative method is converging quickly enough so as not to affect the convergence of the outer solver.

Nsub	Iterations (Solve time (s))		
	S_0^T	S_0^O	S_0^{DO}
2x1	40(100.461)	40(100.454)	39(99.1487)
4x1	39(54.595)	39(52.8131)	39(53.2067)
8x1	51(107.709)	>100(-)	48(101.096)
Nsub	Iterations (solve time (s))		
	S_2^T	S_2^O	S_2^{DO}
2x1	40(104.899)	40 (105.597)	40(104.916)
4x1	43(63.0972)	43(62.7721)	44(63.0429)
8x1	54(119.72)	49(107.059)	97(212.781)

Table 5.13: *2D BP-Eage model, $\omega = 3\pi$. Outer FGMRES solver, exiting when the Euclidean norm of the residual $< 10^{-5}$. The total solve time in seconds is given in brackets*

Nsub	Iterations (Solve time (s))		
	S_0^T	S_0^O	S_0^{DO}
2x2x1	32(592.099)	32(533.992)	32(599.09)
4x4x1	32(171.142)	32(169.913)	32(162.494)
8x8x1	32(63.457)	32(60.9277)	31(60.5457)
Nsub	Iterations (solve time (s))		
	S_2^T	S_2^O	S_2^{DO}
2x2x1	32(637.672)	32(671.308)	32(663.98)
4x4x1	32(195.437)	32(195.287)	32(201.002)
8x8x1	31(70.2567)	32(70.1341)	32(73.4446)

Table 5.14: *3D SEG-Salt model, $\omega = 8\pi$. Outer FGMRES solver, exiting when the Euclidean norm of the residual $< 10^{-6}$. The total solve time in seconds is given in brackets*

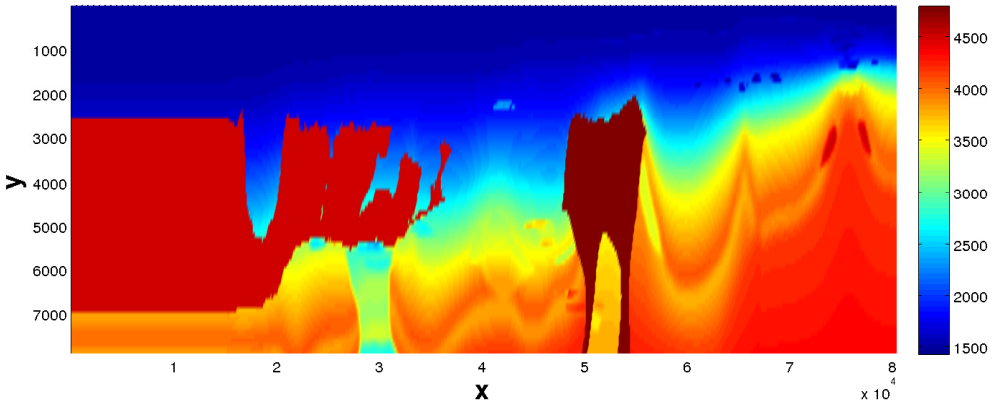

Figure 5-16: *The BP-Eage velocity model.*

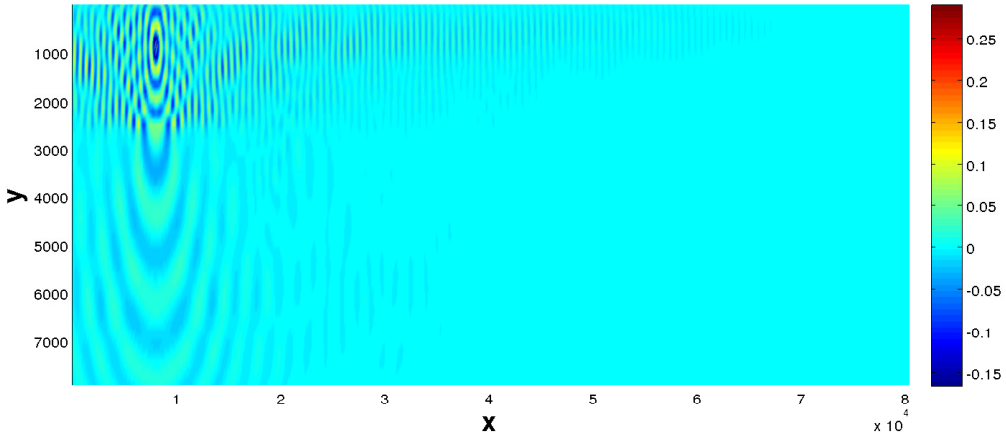

Figure 5-17: *The real part of the numerical solution of the Helmholtz equation with the 2D BP-Eage velocity model with $\omega = 3\pi$.*

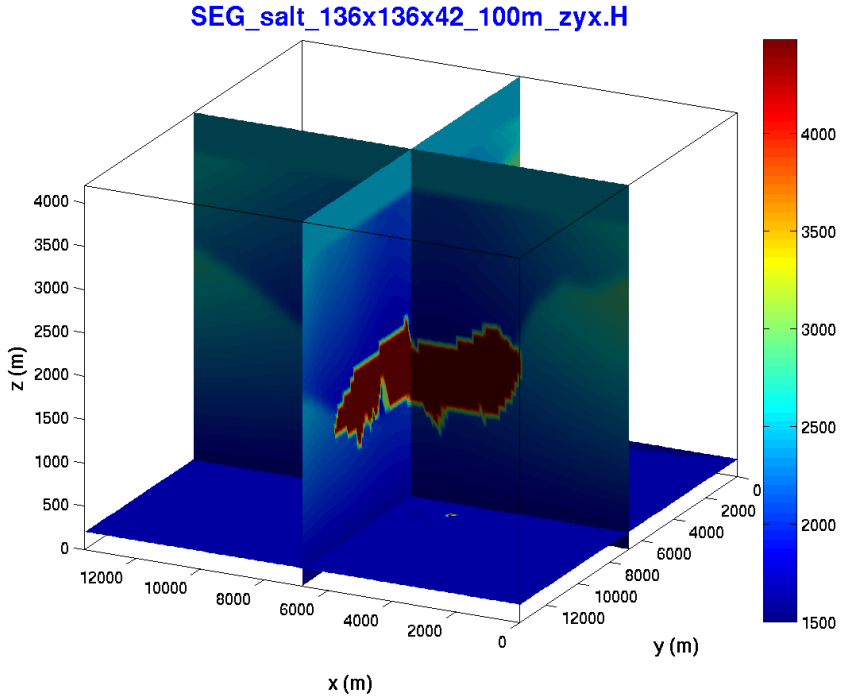

Figure 5-18: *The SEG-Salt velocity model.*

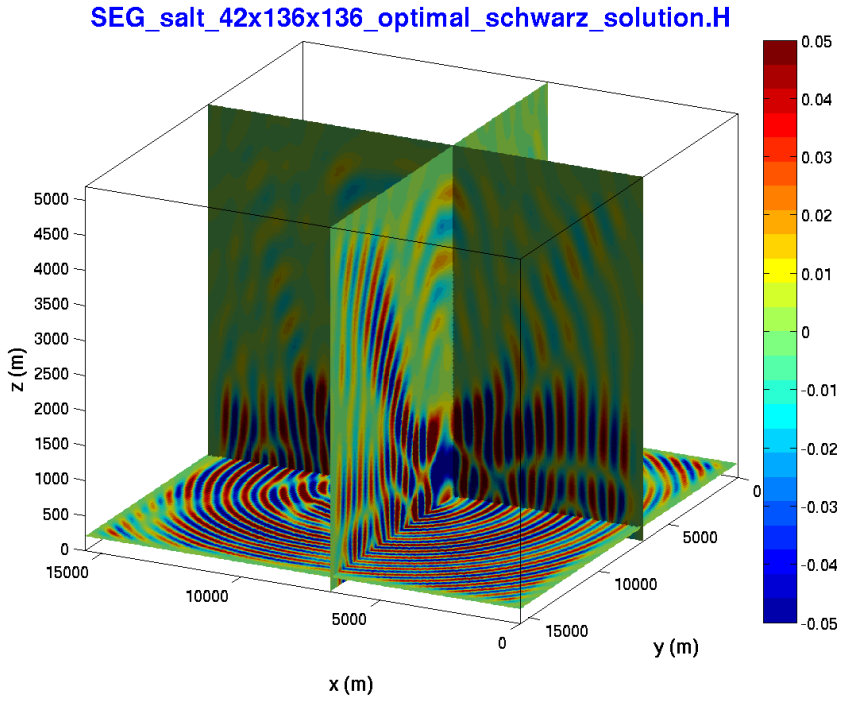

Figure 5-19: *Numerical solution of the Helmholtz equation with the 3D SEG-Salt velocity model with* $\omega = 6\pi$.

5.5.5 Experiments with Hybrid method for increasing ω

We now perform some numerical experiments on a $2D$ and $3D$ industrial problem. Our set up for each run is that outlined in table 5.11 with an interface condition of S_0^{DO}. The purpose of these tests are to examine the performance of the hybrid method for increasing ω. The domain decomposition is such that each subdomain which we solve on is assigned its own core.

The first velocity model that we use is the $2D$ Marmousi velocity model [54] shown in Figure 5-20 with the real part of the numerical solution show in Figure 5-21. We note that the total number of grid points (including PMLs) for this model are $N = 394 \times 135 = 53190$. Our convergence criterion for the outer FGMRES solver is that the Euclidean norm of the residual is less than or equal to 10^{-5}. The source that we use is once again a Gaussian point source located at $\mathbf{x} = (7500, 50)$. We firstly address the performance of the hybrid method as the angular frequency ω is increased. Tables 5.15 ($\omega = 7\pi$), 5.16 ($\omega = 10\pi$) and 5.17 ($\omega = 14\pi$) show the number of iterations and corresponding computational time in seconds. We can observe from these tables that the number of iterations, of the outer FGMRES solver, for a fixed ω is constant as we increase the number of subdomains. When ω increases the total outer iterations of FGMRES increases moderately from 13 iterations when $\omega = 7\pi$ up to 21 iterations when $\omega = 14\pi$. This is clearly not a number of iterations which is independent of ω, but the increase is at least less than linear. This is slightly disappointing as previously we had observed that GMRES preconditioned with the sweeping precondtioner, where the moving PML regions involve solves with direct solvers, converges with a number of iterations which is independent of ω. However the method is completely parallel and does not rely on the use of a parallel direct solver which have high memory requirements.

The $3D$ model that we solve is the SEG-Salt velocity model shown in Figure 5-18, and corresponding numerical solution in Figure 5-19. We fix the problem size to have a total number of grid points $N = 1108432$ and increase the angular frequency, from $\omega = 3\pi$, 6π, 9π. We choose to decompose our total domain into $n_{sub} \times n_{sub} \times 1$ subdomains where $n_{sub} = 2$, 4, 8 for a total number of subdomains of $N_{sub} = 4$, 16, 64. For each different choice of ω we increase the number of subdomains and note the corresponding number of iterations taken by the hybrid method and the corresponding solve time. The source we use is a Gaussian point source located at $\mathbf{x} = (6000, 6000, 50)$ of the form,

$$f(x, y, z) = \exp\left(-\left(\frac{\omega}{\pi}\right)^2 \left[(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2\right]\right)$$

The results are presented in tables 5.18, 5.19 and 5.20. The conclusions that we can draw from the $3D$ numerical results are similar to those in $2D$. Firstly the number of iterations stays constant as the number of subdomains is increased. When ω is doubled from 3π to 6π the number of iterations increases by only three. However when ω is increased to 9π the number of iterations jumps up to 43 iterations. This is a large jump and is due to the fact that we only use around 4 points per wavelength in the discretisation for $\omega = 9\pi$ which is still deemed an acceptable accuracy for some in the Seismic community. When $\omega = 3\pi$ we have around 13 grid points per wavelength and when $\omega = 6\pi$ we have around 7 grid points per wavelength. We expect that this method should show similar ω independent convergence if the appropriate grid points per wavelength were used, however at the time of computing we had limited computational resources.

Nsub	Iterations	Solve time (s)
2x1	13	2.256e+01
4x1	13	1.770e+01
8x1	13	1.370+01
12x1	13	1.420e+01

Table 5.15: *2D Marmousi model with $\omega = 7\pi$.*

Nsub	Iterations	Solve time (s)
2x1	18	3.079e+01
4x1	18	2.274e+01
8x1	18	1.732e+01
12x1	18	1.830e+01

Table 5.16: *2D Marmousi model with $\omega = 10\pi$.*

Nsub	Iterations	Solve time (s)
2x1	21	3.111e+01
4x1	21	2.504e+01
8x1	21	1.946e+01
12x1	21	1.984e+01

Table 5.17: *2D Marmousi model with $\omega = 14\pi$.*

Nsub	Iterations	Solve time (s)
2x2x1	26	2.704e+01
4x4x1	26	2.470e+01
8x8x1	26	9.440e+00

Table 5.18: *3D SEG-Salt model with $\omega = 3\pi$.*

Nsub	Iterations	Solve time (s)
2x2x1	29	2.995e+01
4x4x1	29	2.691e+01
8x8x1	29	1.011e+01

Table 5.19: *3D SEG-Salt model with $\omega = 6\pi$.*

Nsub	Iterations	Solve time (s)
2x2x1	43	9.98e+01
4x4x1	43	9.97e+01
8x8x1	43	9.99e+01

Table 5.20: *3D SEG-Salt model with $\omega = 9\pi$.*

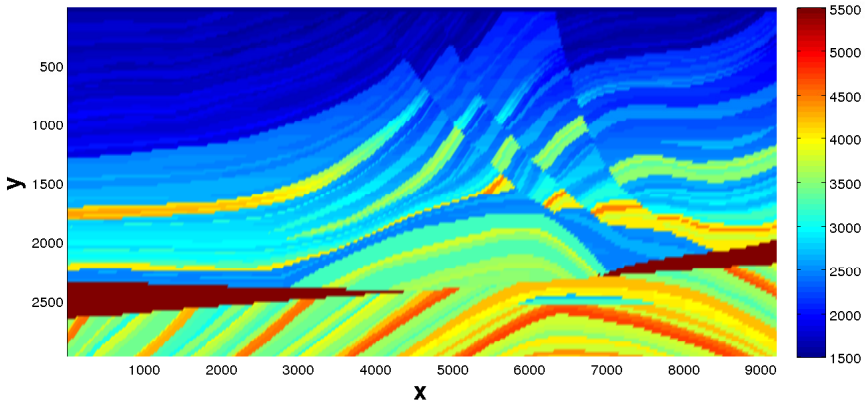

Figure 5-20: *The full Marmousi velocity model.*

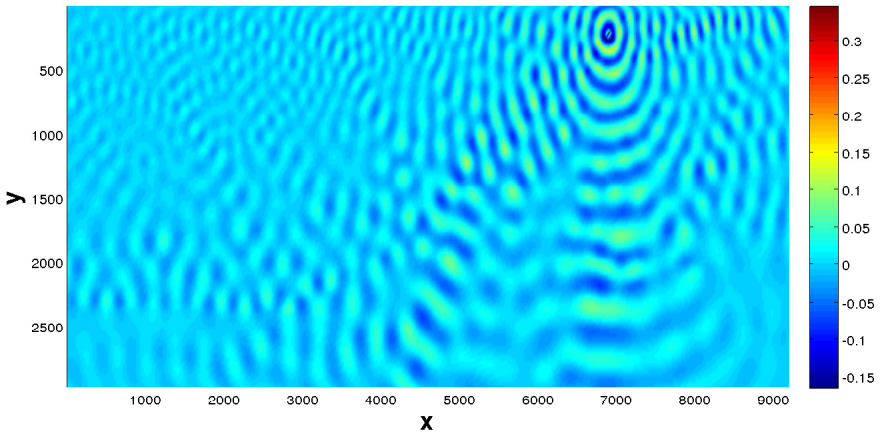

Figure 5-21: *The real part of the numerical solution of the Helmholtz equation with the Marmousi velocity model with* $\omega = 10\pi$.

CHAPTER 6

CONCLUSIONS AND FURTHER WORK

In this thesis we studied and developed iterative methods for solving the linear systems arising from discretisations of the Helmholtz equation. In Chapter 1 contained no original material and gave an overview of our model interior impedance Helmholtz problem and discussed a particular application in Seismic inversion. The model problem was then discretised using low order finite elements to form a linear system of equations. The difficulties in solving this linear system by conventional methods was then discussed.

The first half of chapter 2 served as a review of material from the literature. Initially we reviewed the Generalized Minimal Residual methods (GMRES)[57]. We also provided a short review of the convergence results for GMRES of Elmann and others [2], [17] which provide an upper bound on the residual involving the field of values of the system matrix of the linear system we are solving. This result tells one that if the field of values of our system matrix is bounded away from the origin then we should expect GMRES to converge in a finite number of iterations. It was demonstrated numerically that the field of values of the system matrix arising from a finite element discretisation contained the origin, and hence we should expect poor GMRES convergence which was observed numerically. This then motivated a discussion on preconditioning methods for Helmholtz problems and we introduced the popular *shifted* Laplace preconditioner. To describe this preconditioner we first introduced a PDE problem, similar to our model problem. For this new PDE the wavenumber k^2 in our Helmholtz operator is replaced by $k^2 + i\epsilon$ for $\epsilon > 0$ (where $\epsilon = \mathcal{O}(k^\delta)$ for $\delta \in [0, 2]$), where ϵ acts as an artificial damping. We then computed numerically the field of values of the finite element discretisation of this new PDE problem and demonstrated that when ϵ was increased, up to $\epsilon = k^2$, the distance of the boundary of the field of values from

the origin increases. The linear system was then solved using GMRES and the best convergence was observed when $\epsilon = k^2$. Recent work by Gander, Graham and Spence [38] on how to choose this value of ϵ when using a *shifted* Laplace preconditioner was then discussed. The second half of Chapter 2 was concerned with the Schwarz method, a domain decomposition method. Initially we reviewed previous authors work on Schwarz methods for the Helmholtz problem, however everything that followed was original material. We then considered solving the complex shifted Helmholtz problem iteratively using a Schwarz method. What followed was then a Fourier analysis of the two subdomain Multiplicative overlapping Schwarz method for the solution of the complex shifted Helmholtz problem. This Fourier analysis resulted in the calculation of a convergence rate for the Schwarz algorithm in terms of k, ϵ, ξ (the Fourier frequency) and S which was the choice of operator on the subdomain interfaces. We then showed that choosing $S = ik$ (which results in an impedance interface condition) improved the asymptotic behaviour of the convergence rate compared to the classical choice of a Dirichlet boundary condition. For example we showed that when the subdomains overlap, that the convergence rate of the classical algorithm behaves like $1 - k^{\delta-2}$ for $k \to \infty$, compared to that the algorithm with impedance condition which behaves like $1 - k^{\frac{\delta-2}{2}}$. Furthermore choosing an impedance condition on the interface actually results in an iterative algorithm which converges without overlap in the subdomains, which is a requirement for the classical algorithm. We also discussed the inclusion of a second order term in the interface condition.

In Chapter 3 all of the work is original. We used the novel approach of choosing $S = p(1 + i)$ where $p \in \mathbb{R}^+$ and then trying to use p to minimise the maximum of our convergence rate of the non-overlapping optimised Schwarz algorithm. The resulting minimax problem was then solved analytically using asymptotic methods, resulting in a closed form expression for p in terms of k and ϵ. It was then shown that using this new interface condition resulted in a convergence rate which behaves like $1 - k^{\frac{\delta-2}{4}}$ for $k \to \infty$, an improvement over the standard impedance interface condition. This then allowed us to derive a lower bound on the number of Schwarz iterations for our iterative algorithm in terms of k and ϵ. Furthermore this lower bound showed us that if $\epsilon = k^2$ then we expect the Schwarz algorithm to converge in a number of iterations which is independent of k.

In Chapter 4 we performed numerical experiments using the methods discussed in Chapters 2 and 3. The first set of experiments investigated the overlapping Schwarz algorithm and non-overlapping Schwarz algorithm with an additional second order term in the interface condition. As obtaining a closed form solution for both of these problems is technically very involved we solved both of the problems numerically and conjectured the resulting behaviour for $k \to \infty$. The conjectured behaviour showed that

we should expect improved convergence when including overlap (namely asymptotic behaviour of the convergence rate of $1 - k^{\frac{\delta-2}{6}}$ as k increases), or second order terms without overlap (asymptotic behaviour of the convergence rate of $1-k^{\frac{\delta-2}{8}}$ as k increases) in the interface condition. Both of these conjectured results are original contributions. In the second set of numerical experiments we implemented the two domain Schwarz method as both an iterative solver and as a preconditioner for GMRES for the solution of the complex shifted Helmholtz problem. We then tested the convergence of the iterative algorithm and preconditioned GMRES using all of the interface conditions mentioned in Chapter 2 and 3 in the Schwarz algorithm. The numerical results agreed well with the convergence results presented previously, and when $\epsilon = k^2$ convergence independent of k was observed as expected. Finally we looked at using GMRES to solve our original Helmholtz problem preconditioned with a domain decomposition method (with multiple subdomains) involving solves with the Helmholtz problem with a complex shift. The goal of these experiments was to examine how one should choose ϵ in the preconditioner to obtain the best convergence for GMRES. It was observed that a choice of $\epsilon \sim k$ resulted in the lowest number of GMRES iterations. Furthermore computations of the field of values of the preconditioned system using a two subdomain decomposition showed that when $\epsilon \sim k$ the field of values was bounded away from the origin.

In Chapter 5 we developed a new hybrid preconditioner for the solution of large scale Helmholtz problems by combining the domain decomposition methods discussed previously with the *sweeping* preconditioner of Engquist and Ying [5]. We started by giving a review of the sweeping preconditioner and PML boundary conditions which are essential to the algorithm. This was followed by some model computations which showed that a choice of $\epsilon = k$ in the sweeping preconditioner resulted in the best performance for GMRES. This was then followed by the main original contribution of this chapter. We initially introduced the new hybrid method which replaces the direct solves in the sweeping preconditioner with an iterative solver preconditioned with a domain decomposition method (with multiple domains). This new method was then tested on some challenging industrial model problems in 2D and 3D. However, there still remains further work to make the method fully scalable in 3D.

Finally we suggest some possible directions for further work arising from this thesis:

- For overlapping optimised Schwarz methods and those where a second order term is included in the interface condition, further work should be done to establish closed form expressions for the optimised interface conditions.

- Currently the analysis presented in Chapters 2, 3 is only for a two subdomain

decomposition, this should be extended to the multi subdomain case.

- Further work (which is being looked into by Paul Childs at Schlumberger) should be done to make the hybrid preconditioner of Chapter 5 scalable in 3D. A possible idea is the inclusion of a coarse space correction in the domain decomposition method.

REFERENCES

[1] E.Turkel A. Bayliss, C.I. Goldstein. An iterative method for the Helmholtz equation. *Journal of Computational Physics*, 49(3):443–457, 1983.

[2] E.E. Tyrtyshnikov B. Beckerman, S.A Goreinov. Some remarks on the elman estimate for GMRES. *SIAM J Matrix Anal. Appl.*, 27(3):772–778, 2006.

[3] P. Joly B. Després and J. E. Roberts. A domain decomposition method for the harmonic Maxwell's equations. *IMACS International Symposium on Iterative Methods in Linear Algebra, North-Holland, Amsterdam*, pages 475–484, 1992.

[4] H-K. Zhao B. Engquist. Absorbing boundary conditions for domain decomposition. *Applied Numerical Mathematics*, 27(4):341–365, 1998.

[5] L. Ying B. Engquist. Sweeping preconditioner for the Helmholtz equation: moving perfectly matched layers. *SIAM Multiscale Modelling and Simulation*, 9(-):686–710, 2010.

[6] L. Ying B. Engquist. Sweeping preconditioner for the Helmholtz equation: Hierarchical matrix representation. *Communications on Pure and Applied Mathematics*, 64(-):697–735, 2011.

[7] J. Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *J. Computational Physics*, 114(2):185–200, 1994.

[8] P.E. Bjørstad B.F. Smith and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, Cambridge, UK, 1st edition, 1996.

[9] X.-C. Cai and M. Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM journal on scientific computing*, 21:792–797, 1999.

[10] X.-C. Cai and O.B. Widlund. Domain decomposition for indefinite elliptic problems. *SIAM J. Sci. Statist. Comput*, 13:243–258, 1992.

[11] O. Cessenat and B. Després. Application of an ultra weak variational formulation of elliptic pdes to the two-dimensional helmholtz problem. *SIAM J. Numer. Anal.*, 35:225–299, 1998.

[12] D. L. Colton and R. Kress. *Integral Equation Methods in Scattering Theory*. Wiley, 1983.

[13] J.W. Cooley and J.W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Math. Comp*, 19:297–301, 1965.

[14] C.C. Cowen and E. Harel. An effective algorithm for computing the numerical range. *Unpublished manuscript*, 1995.

[15] B. Després. Domain decomposition and the Helmholtz problem, ii. *Proceedings of the second international conference on mathematical and numerical aspects of wave propagation, Newark*, pages 197–206, 1992.

[16] H. Doyle. *Seismology*. John Wiley & Sons Ltd, 3rd edition, 1995.

[17] H. C. Elman. *Iterative methods for sparse nonsymmetric systems of linear equations*. PhD thesis, Yale University, 1982.

[18] B. Engquist and A. Majda. Absorbing boundary conditions for the numerical simulation of waves. *Math. Comp*, 31:629–651, 1977.

[19] J. Brac F. Aminzadeh and T. Kunz. 3D salt and overthrust models. *SEG/EAGE 3D Modeling Series No.1. SEG, Tulsa*, 1997.

[20] C. Tsogka F. Collino. Application of the PML absorbing layer model to the linear elastodynamic problem in anisotropic heterogeneous media. *Geophysics*, 66(1):294 – 307, 2001.

[21] M.J. Gander. Optimized Schwarz methods for Helmholtz problems. *Proceedings of the Thirteenth International Conference on Domain Decomposition Methods*, pages 245–252, 2001.

[22] M.J. Gander and H. Zhang. Optimized Schwarz methods with overlap for the Helmholtz equation. *Domain Decomposition Methods in Science and Engineering XXI, Springer-Verlag*, pages 181–189, 2013.

[23] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

[24] A. Greenbaum. *Iterative Methods for Solving Linear Systems*. SIAM, 1997.

[25] S. A. Sauter I. M. Babuska. Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers? *SIAM Rev*, 42(3):451–484, 2000.

[26] J. K. Reid I. S. Duff. The multifrontal solution of indefinite sparse symmetric linear. *ACM Transactions on Mathematical Software (TOMS)*, 9(3):302–325, 1983.

[27] E. Turkel I. Singer. High-order finite difference methods for the Helmholtz equation. *Comput. Methods Appl. Mech. Engrg*, 163(-):343–358, 1998.

[28] F. Ihlenberg and I.Babuška. Finite element solution of the Helmholtz equation with high wave number part i: The h-version of the FEM. *Computers & Mathematics with Applications*, 30(9):9–37, 1995.

[29] F. Ihlenburg. *Finite element analysis of acoustic scattering*. Springer, 1998.

[30] B. Després J.-D. Benamou. A domain decomposition method for the Helmholtz equation and related optimal control problems. *J. Comput. Phys*, 136:68–82, 1997.

[31] M. Sarkis J-H. Kimm. Shifted Laplacian RAS solvers for the Helmholtz equation. *Proceedings of the 20th international conference on Domain Decomposition methods*, 2011.

[32] S. Li J. Poulson, B. Engquist and L. Ying. A parallel sweeping preconditioner for heterogeneous 3D Helmholtz equations. *SIAM Journal on Scientific Computing*, 35(3):194–212, 2012.

[33] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. SIAM, 1995.

[34] R. Krause L. Conen, V. Dolean and F. Nata. A coarse space for heterogeneous Helmholtz problems based on the Dirichlet-to-Neumann operator. *Journal of Computational and Applied Mathematics*, 271:83–99, 2014.

[35] P.-L. Lions. On the Schwarz alternating method. iii. *Proceedings of the third international symposium on domain decomposition methods*, pages 202–223, 1990.

[36] J. W. Longley. Modified Gram-Schmidt process vs. classical Gram-Schmidt. *Communications in Statistics - Simulation and Computation*, 10(5):517–527, 1981.

[37] J. M. Melenk. *On generalized finite element methods*. PhD thesis, The University of Maryland, 1995.

[38] E.A. Spence M.J. Gander, I.G. Graham. How should one choose the shift for the shifted Laplacian to be a good preconditioner for the Helmholtz equation? *Preprint*, pages 1–34, 2014.

[39] F. Nataf M.J. Gander, F. Magoulés. Optimized Schwarz methods without overlap for the Helmholtz equation. *SIAM J. Sci. Comput*, 24(1):38–60, 2002.

[40] L. Halpern F. Magoulés M.J. Gander. An optimized Schwarz method with two-sided robin transmission conditions for the Helmholtz equation. *Int. J. Numer. Meth. Fluids*, 55(1):163–175, 2007.

[41] A. Moiola and E.A. Spence. Is the helmholtz equation really sign-indefinite? *SIAM Review*, 56(2):274–312, 2014.

[42] J.A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(6):308–313, 1965.

[43] Picture of seismic data acquisition. http://.krisenergy.com/company/about-oil-and-gas/exploration. Accessed: 2014-11-24.

[44] M.J. Gander O.G. Ernst. Why is it difficult to solve Helmholtz problems with classical iterative methods. *Numerical analysis of multiscale problems*, 83(-):325–363, 2012.

[45] V. Druskin P. N. Childs and L. Knizhnerman. Preconditioning the helmholtz equation using approximate dirichlet-to-neumann operators on optimal grids. *IMA conference on Numerical Linear Algebra, Birmingham*, 2014.

[46] I.G. Graham P.N. Childs and J.D. Shanks. Hybrid sweeping preconditioners for the Helmholtz equation. *Proc. 11th conference on mathematical and numerical aspects of wave propagation (Gammarth, Tunisia, June 2013)*, pages 285–286, 2013.

[47] J. Poulson. *Fast Parallel Solution of Heterogeneous 3D Time-harmonic Wave Equations*. PhD thesis, University of Texas as Austin, 2012.

[48] R. Gerhard Pratt. Seismic waveform inversion in the frequency domain, part 1: Theory and verification in a physical scale model. *Geophysics*, 64(3):888–901, 1999.

[49] R. Gerhard Pratt. Seismic imaging of complex onshore structures by two-dimensional elastic frequency-domain full-waveform inversion. *Geophysics*, 74(6):105–118, 2009.

[50] Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM Journal on Scientific Computing*, 14(2):461–469, 1993.

[51] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, 2nd edition, 1996.

[52] H. A. Schwarz. Über einen grenzübergang durch alternierendes verfahren (On passing to the limit by the alternating process). *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich.*, 15:272–286, 1870.

[53] L. H Thomas. Elliptic problems in linear difference equations over a network. *Watson Sci Comput. Lab. Rept. Columbia University, New York*, 1949.

[54] R.J. Versteeg. The Marmousi experience: Velocity model determination on a synthetic complex data set. *The Leading Edge*, 1994.

[55] J. Virieux and S. Operto. An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74(6):126–152, 2009.

[56] D. Watkins. *Fundamentals of Matrix Computations.* Wiley, 3rd edition, 2010.

[57] M. Schultz Y. Saad. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comp*, 7(3):856–869, 1986.

[58] C. Vuik Y.A. Erlangga, C.W. Oosterlee. Comparison of Multigrid and incomplete ILU shifted-Laplace preconditioners for the inhomogeneous Helmholtz equation. *Applied Numerical Mathematics*, 56(5):648–666, 2006.

[59] C. Vuik Y.A. Erlangga, C.W. Oosterlee. A novel multigrid based preconditioner for heterogeneous Helmholtz problems. *SIAM J. Sci. Comp*, 27(-):1471–1492, 2006.